

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2005-346005

(43)Date of publication of application : 15.12.2005

(51)Int.Cl. G09C 1/00

(21)Application number : 2004-168997 (71)Applicant : NTT COMMUNICATIONS KK

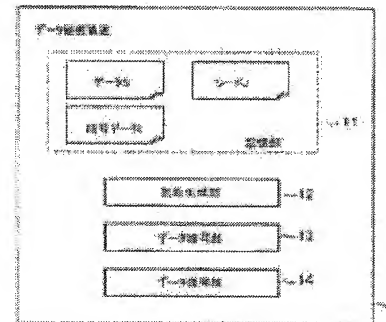
(22)Date of filing : 07.06.2004 (72)Inventor : OGIWARA TOSHIHIKO
KAGAYA MAKOTO
NOMURA SUSUMU

(54) DEVICE, METHOD, AND PROGRAM FOR CONCEALING DATA

(57)Abstract:

PROBLEM TO BE SOLVED: To conceal data for a long period of time in Vernam cipher and other cipher systems, wherein data is enciphered by calculating an exclusive OR of a random number sequence generated by a pseudo-random number generating algorithm using a block cipher and the data to be concealed.

SOLUTION: This data concealing device 1 comprises a storage section 11, a random number generating section 12, a data ciphering section 13, and a data deciphering section 14, and conceals data by using Vernam cipher. The random number generating section 12 generates a random number K(J) from a seed J, by using a pseudo-random number algorithm K (K=F, G, H, ...) having the block cipher as its component. The data ciphering section 13 calculates exclusive OR (XOR) of the data S and the random number K(J) and generates cipher data I. When the block cipher is predicted to becoming vulnerable, the ciphering section 13 calculates exclusive OR (XOR) of the cipher data I and the random number K (J), and further, enciphers the cipher data I to generate cipher data I'.



(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-346005

(P2005-346005A)

(43) 公開日 平成17年12月15日(2005. 12. 15)

(51) Int.Cl.⁷
G09C 1/00

F I

G09C 1/00 610Z
G09C 1/00 650B
G09C 1/00 650Z

テーマコード(参考)

5J104

審査請求 未請求 請求項の数 20 O L (全 40 頁)

(21) 出願番号 特願2004-168997(P2004-168997)
(22) 出願日 平成16年6月7日(2004.6.7)

(71) 出願人 399035766
エヌ・ティ・ティ・コミュニケーションズ
株式会社
東京都千代田区内幸町一丁目1番6号
(74) 代理人 100083806
弁理士 三好 秀和
(74) 代理人 100095500
弁理士 伊藤 正和
(74) 代理人 100101247
弁理士 高橋 俊一
(74) 代理人 100098327
弁理士 高松 俊雄

最終頁に続く

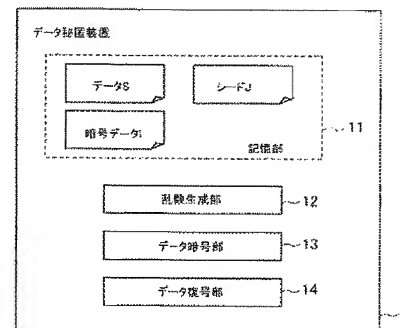
(54) 【発明の名称】 データ秘匿装置、データ秘匿方法、及びデータ秘匿プログラム

(57) 【要約】

【課題】バーナム暗号をはじめとする、ブロック暗号を用いた擬似乱数生成アルゴリズムにより生成された乱数列と、秘匿したいデータとの排他的論理和を計算してデータを暗号化をする暗号化方式において、長期間にわたってデータを秘匿することができる。

【解決手段】データ秘匿装置1は、記憶部11、乱数生成部12、データ暗号部13、及びデータ復号部14を備え、バーナム暗号を用いてデータを秘匿する。乱数生成部12は、ブロック暗号を構成要素に持つ擬似乱数アルゴリズムK(K=F, G, H, ...)を用いてシードJから乱数K(J)を生成する。データ暗号部13は、データSと、乱数K(J)との排他的論理和(XOR)を計算して、暗号データIを生成する。また、ブロック暗号の脆弱化が予測されたときは、暗号データIと乱数K(J)との排他的論理和(XOR)を計算して、暗号データIをさらに暗号化した暗号データI'を生成する。

【選択図】図1



【特許請求の範囲】

【請求項1】

データをバーナム暗号を用いて秘匿するデータ秘匿装置であって、
ブロック暗号を構成要素に持つ擬似乱数生成関数を用いて第1の乱数を生成する手段と

、
前記データと、前記第1の乱数との排他的論理和により暗号データを生成する手段と、
生成された暗号データを所定の記憶部に記憶する手段と、

所定の時期に、前記記憶部に記憶された暗号データの生成時に用いられた擬似乱数生成関数よりも計算量が大きい別の擬似乱数生成関数を用いて、第2の乱数を生成する手段と

、
前記記憶部に記憶された暗号データと、前記第2の乱数との排他的論理和により暗号データを生成する暗号データ暗号化手段と、

前記記憶部に記憶された暗号データに代えて、前記暗号データ暗号化手段で生成された暗号データを前記記憶部に記憶する手段と、
を有することを特徴とするデータ秘匿装置。

【請求項2】

前記暗号データ暗号化手段は、さらに、生成された暗号データと、前記記憶部に記憶された暗号データの生成時に用いられた乱数との排他的論理和により、暗号データを生成することを特徴とする請求項1記載のデータ秘匿装置。

【請求項3】

前記記憶部に記憶された暗号データと、前記データから前記記憶部に記憶された暗号データを生成するまでに用いた乱数すべてとを結合した排他的論理和により、前記データを復号する手段を有することを特徴とする請求項1記載のデータ秘匿装置。

【請求項4】

前記記憶部に記憶された暗号データと、前記記憶部に記憶された暗号データを生成するときに用いた乱数との排他的論理和により、前記データを復号する手段を有することを特徴とする請求項2記載のデータ秘匿装置。

【請求項5】

データを秘密分散法を用いて秘匿するデータ秘匿装置であって、

前記秘密分散法は、

前記データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割方法であり、前記データを処理単位ビット長毎に区分けして、複数の元部分データを生成し、この複数の元部分データの各々に対応して、前記データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成して、所望の分割数の分割データを生成するとともに、

新たに発生させた乱数から処理単位ビット長の複数の乱数部分データを生成し、前記各分割部分データと該乱数部分データの排他的論理和により処理単位ビット長毎に再分割部分データを生成して、前記所望の分割数の再分割データを生成するデータ分割方法であり

、
ブロック暗号を構成要素に持つ擬似乱数生成関数を用いて第1の乱数を生成する手段と

、
前記データと前記第1の乱数から、前記秘密分散法を用いて複数の分割データを生成する手段と、

前記複数の分割データを暗号データとして所定の記憶部に記憶する手段と、

所定の時期に、前記記憶部に記憶された暗号データの生成時に用いられた擬似乱数生成関数よりも計算量が大きい別の擬似乱数生成関数を用いて、第2の乱数を生成する手段と

前記記憶部に記憶された暗号データと前記第2の乱数から、前記秘密分散法を用いて新たな分割データを生成する手段と、

前記記憶部に記憶された暗号データに代えて、前記新たな分割データを暗号データとして前記記憶部に記憶する手段と、
を有することを特徴とするデータ秘匿装置。

【請求項6】

前記記憶部に記憶された暗号データのうち、復元可能な所定の個数の分割データの組み合わせから、前記秘密分散法を用いて、前記データを復号する手段を有することを特徴とする請求項5記載のデータ秘匿装置。

【請求項7】

前記秘密分散法は、前記各分割部分データの定義式における乱数部分データを、該乱数部分データと該各乱数部分データに対応する新たな乱数部分データとの排他的論理和演算に置換した各再分割部分データの定義式により、各再分割部分データを生成することを特徴とする請求項5又は6記載のデータ秘匿装置。

【請求項8】

前記秘密分散法は、前記各分割部分データの定義式における乱数部分データを、該乱数部分データに対応する新たな乱数部分データに置換した各再分割部分データの定義式により、各再分割部分データを生成することを特徴とする請求項5又は6記載のデータ秘匿装置。

【請求項9】

前記所定の時期は、コンピュータの計算能力をもとに判断された、前記擬似乱数関数に用いられたブロック暗号の脆弱時期であることを特徴とする請求項1乃至8のいずれか1項に記載のデータ秘匿装置。

【請求項10】

データをバーナム暗号を用いて秘匿するデータ秘匿方法であって、
ブロック暗号を構成要素に持つ擬似乱数生成関数を用いて第1の乱数を生成するステップと、
前記データと、前記第1の乱数との排他的論理和により暗号データを生成するステップと、
生成された暗号データを所定の記憶部に記憶するステップと、
所定の時期に、前記記憶部に記憶された暗号データの生成時に用いられた擬似乱数生成関数よりも計算量が大きい別の擬似乱数生成関数を用いて、第2の乱数を生成するステップと、
前記記憶部に記憶された暗号データと、前記第2の乱数との排他的論理和により暗号データを生成する暗号データ暗号化ステップと、
前記記憶部に記憶された暗号データに代えて、前記暗号データ暗号化ステップで生成された暗号データを前記記憶部に記憶するステップと、
を有することを特徴とするデータ秘匿方法。

【請求項11】

データを秘密分散法を用いて秘匿するデータ秘匿方法であって、
前記秘密分散法は、
前記データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割方法であり、前記データを処理単位ビット長毎に区分けして、複数の元部分データを生成し、この複数の元部分データの各々に対応して、前記データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成して、所望の分割数の分割データを生成するとともに、

新たに発生させた乱数から処理単位ビット長の複数の乱数部分データを生成し、前記各分割部分データと該乱数部分データの排他的論理和により処理単位ビット長毎に再分割部

分データを生成して、前記所望の分割数の再分割データを生成するデータ分割方法であり、

ブロック暗号を構成要素に持つ擬似乱数生成関数を用いて第1の乱数を生成するステップと、

前記データと前記第1の乱数から、前記秘密分散法を用いて複数の分割データを生成するステップと、

前記複数の分割データを暗号データとして所定の記憶部に記憶するステップと、

所定の時期に、前記記憶部に記憶された暗号データの生成時に用いられた擬似乱数生成関数よりも計算量大きい別の擬似乱数生成関数を用いて、第2の乱数を生成するステップと、

前記記憶部に記憶された暗号データと前記第2の乱数から、前記秘密分散法を用いて新たな分割データを生成するステップと、

前記記憶部に記憶された暗号データに代えて、前記新たな分割データを暗号データとして前記記憶部に記憶するステップと、
を有することを特徴とするデータ秘匿方法。

【請求項12】

データをバーナム暗号を用いて秘匿するためのコンピュータが読み取り可能なデータ秘匿プログラムであって、

ブロック暗号を構成要素に持つ擬似乱数生成関数を用いて第1の乱数を生成するステップと、

前記データと、前記第1の乱数との排他的論理和により暗号データを生成するステップと、

生成された暗号データを所定の記憶部に記憶するステップと、

所定の時期に、前記記憶部に記憶された暗号データの生成時に用いられた擬似乱数生成関数よりも計算量大きい別の擬似乱数生成関数を用いて、第2の乱数を生成するステップと、

前記記憶部に記憶された暗号データと、前記第2の乱数との排他的論理和により暗号データを生成する暗号データ暗号化ステップと、

前記記憶部に記憶された暗号データに代えて、前記暗号データ暗号化ステップで生成された暗号データを前記記憶部に記憶するステップと、
を前記コンピュータに実行させることを特徴とするデータ秘匿プログラム。

【請求項13】

前記暗号データ暗号化ステップは、さらに、生成された暗号データと、前記記憶部に記憶された暗号データの生成時に用いられた乱数との排他的論理和により、暗号データを生成することを特徴とする請求項12記載のデータ秘匿プログラム。

【請求項14】

前記記憶部に記憶された暗号データと、前記データから前記記憶部に記憶された暗号データを生成するまでに用いた乱数すべてとを結合した排他的論理和により、前記データを復号するステップを前記コンピュータに実行させることを特徴とする請求項12記載のデータ秘匿プログラム。

【請求項15】

前記記憶部に記憶された暗号データと、前記記憶部に記憶された暗号データを生成するときに用いた乱数との排他的論理和により、前記データを復号するステップを有することを特徴とする請求項13記載のデータ秘匿プログラム。

【請求項16】

データを秘密分散法を用いて秘匿するためのコンピュータが読み取り可能なデータ秘匿プログラムであって、

前記秘密分散法は、

前記データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割方法であり、前記データを処理単位ビット長毎に区分けして、複数の元部分デ

ータを生成し、この複数の元部分データの各々に対応して、前記データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成して、所望の分割数の分割データを生成するとともに、

新たに発生させた乱数から処理単位ビット長の複数の乱数部分データを生成し、前記各分割部分データと該乱数部分データの排他的論理和により処理単位ビット長毎に再分割部分データを生成して、前記所望の分割数の再分割データを生成するデータ分割方法であり、

ブロック暗号を構成要素に持つ擬似乱数生成関数を用いて第1の乱数を生成するステップと、

前記データと前記第1の乱数から、前記秘密分散法を用いて複数の分割データに分割するステップと、

前記複数の分割データを暗号データとして所定の記憶部に記憶するステップと、

所定の時期に、前記記憶部に記憶された暗号データの生成時に用いられた擬似乱数生成関数よりも計算量が大きい別の擬似乱数生成関数を用いて、第2の乱数を生成するステップと、

前記記憶部に記憶された暗号データと前記第2の乱数から、前記秘密分散法を用いて新たな分割データを生成するステップと、

前記記憶部に記憶された暗号データに代えて、前記新たな分割データを暗号データとして前記記憶部に記憶するステップと、
を前記コンピュータに実行させることを特徴とするデータ秘匿プログラム。

【請求項17】

前記記憶部に記憶された暗号データのうち、復元可能な所定の個数の分割データの組み合わせから、前記秘密分散法を用いて、前記データを復号するステップを前記コンピュータに実行させることを特徴とする請求項16記載のデータ秘匿プログラム。

【請求項18】

前記秘密分散法は、前記各分割部分データの定義式における乱数部分データを、該乱数部分データと該各乱数部分データに対応する新たな乱数部分データとの排他的論理和演算に置換した各再分割部分データの定義式により、各再分割部分データを生成することを特徴とする請求項16又は17記載のデータ秘匿プログラム。

【請求項19】

前記秘密分散法は、前記各分割部分データの定義式における乱数部分データを、該乱数部分データに対応する新たな乱数部分データに置換した各再分割部分データの定義式により、各再分割部分データを生成することを特徴とする請求項16又は17記載のデータ秘匿プログラム。

【請求項20】

前記所定の時期は、コンピュータの計算能力をもとに判断された、前記擬似乱数関数に用いられたブロック暗号の脆弱時期であることを特徴とする請求項12乃至19のいずれか1項に記載のデータ秘匿プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、暗号化された情報の解読を防止する技術に関し、特に、ブロック暗号を用いた擬似乱数アルゴリズムにより生成された暗号化用乱数が脆弱化したときに好適に適用できるデータ秘匿装置、データ秘匿方法、及びデータ秘匿プログラムに関する。

【背景技術】

【0002】

データの暗号化方法の1つに、バーナム暗号がある。バーナム暗号とは、データと同じ長さの乱数列を用意し、暗号化に際しては、データのnビット目と乱数列のnビット目の

排他的論理和 (XOR) を計算し、復号化に際しては、暗号化されたデータの n ビット目と乱数列の n ビット目の排他的論理和 (XOR) を計算するものである。

【0003】

以下、排他的論理和演算 (XOR) は、「 $*$ 」なる演算記号で表すことにするが、この排他的論理和演算のビット毎の演算規則での各演算結果は下記のとおりになっている。

【0004】

$0 * 0$ の演算結果は 0

$0 * 1$ の演算結果は 1

$1 * 0$ の演算結果は 1

$1 * 1$ の演算結果は 0

また、XOR演算は交換法則、結合法則が成り立つ。すなわち、

$a*b=b*a$

$(a*b)*c=a*(b*c)$ が成り立つことが数学的に証明される。

【0005】

また、 $a*a=0$ 、 $a*0=0*a=a$ が成り立つ。ここで a, b, c は同じ長さのビット列を表し、 0 はこれらと同じ長さですべて「 0 」からなるビット列を表す。

【0006】

図1 4 に簡単なバーナム暗号の例を示す。平文 X と鍵 Y との排他的論理和演算 (XOR) により、計算された暗号文 Z は、再度、鍵 Y との排他的論理和演算 (XOR) により、平文 X に復号されていることがわかる。

【0007】

一方、ブロック暗号は、データを一定のブロック長に区切って、ブロック単位に暗号鍵を用いて暗号化処理を行う方法であるが、このブロック暗号は、コンピュータを用いても計算量的に解読困難であることによりデータを秘匿する。これに対して、上述したバーナム暗号は、いくら計算リソースがあっても鍵となる乱数列がなければデータを得ることができないものである。しかし、バーナム暗号においては、データと同じ長さの乱数列が必要であるため、シード (乱数生成の種となる情報) を入力として長い乱数列を生成する擬似乱数生成アルゴリズムが用いられることが多い。

【0008】

尚、この出願に関連する先行技術文献情報としては、次のものがある。

【非特許文献1】Information Security Laboratory、“One-Time Pad or Vernam Cipher”、[Online]、[平成16年5月20日検索]、インターネット<URL: <http://islab.oregonstate.edu/koc/ece679/notes/onetime.pdf>>

【非特許文献2】情報処理振興事業協会 通信・放送機構、“暗号技術評価報告書(2002年度版) CRYPTREC Report 2002”、[Online]、[平成16年5月20日検索]、インターネット<URL: http://www.shiba.tao.go.jp/kenkyu/CRYPTREC/PDF/c02_report.pdf>

【発明の開示】

【発明が解決しようとする課題】

【0009】

ところで、既存の擬似乱数生成アルゴリズムの多くは、その構成の一部にブロック暗号を用いている。ブロック暗号は、上述したように計算量的に解読困難であることによりデータを秘匿するものであるから、コンピュータの性能向上とともに脆弱化する。ブロック暗号の脆弱化は、必ずしも擬似乱数生成アルゴリズムの脆弱化につながるわけではないが、ある一部の乱数列から他の部分の乱数列が予測されることもあり得る。従って、既存の擬似乱数アルゴリズムにより生成された乱数列を用いたバーナム暗号は、長期間にわたって秘匿したいデータの暗号化には有効でないという課題がある。

【0010】

本発明は、上記の課題を解決するためになされたものであり、バーナム暗号をはじめとする、ブロック暗号を用いた擬似乱数生成アルゴリズムにより生成された乱数列と、秘匿したいデータとの排他的論理和を計算してデータを暗号化する暗号化方式において、長

期間にわたってデータを秘匿し得るデータ秘匿装置、データ秘匿方法、及びデータ秘匿プログラムを提供することを目的とする。

【課題を解決するための手段】

【0011】

上記目的を達成するため、請求項1記載の本発明は、データをバーナム暗号を用いて秘匿するデータ秘匿装置であって、ブロック暗号を構成要素に持つ擬似乱数生成関数を用いて第1の乱数を生成する手段と、前記データと、前記第1の乱数との排他的論理和により暗号データを生成する手段と、生成された暗号データを所定の記憶部に記憶する手段と、所定の時期に、前記記憶部に記憶された暗号データの生成時に用いられた擬似乱数生成関数よりも計算量大きい別の擬似乱数生成関数を用いて、第2の乱数を生成する手段と、前記記憶部に記憶された暗号データと、前記第2の乱数との排他的論理和により暗号データを生成する暗号データ暗号化手段と、前記記憶部に記憶された暗号データに代えて、前記暗号データ暗号化手段で生成された暗号データを前記記憶部に記憶する手段と、を有することを特徴とする。

【0012】

請求項2記載の本発明は、請求項1記載の発明において、前記暗号データ暗号化手段は、さらに、生成された暗号データと、前記記憶部に記憶された暗号データの生成時に用いられた乱数との排他的論理和により、暗号データを生成することを特徴とする。

【0013】

請求項3記載の本発明は、請求項1記載の発明において、前記記憶部に記憶された暗号データと、前記データから前記記憶部に記憶された暗号データを生成するまでに用いた乱数すべてとを結合した排他的論理和により、前記データを復号する手段を有することを特徴とする。

【0014】

請求項4記載の本発明は、請求項2記載の発明において、前記記憶部に記憶された暗号データと、前記記憶部に記憶された暗号データを生成するときに用いた乱数との排他的論理和により、前記データを復号する手段を有することを特徴とする。

【0015】

請求項5記載の本発明は、データを秘密分散法を用いて秘匿するデータ秘匿装置であって、前記秘密分散法は、前記データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割方法であり、前記データを処理単位ビット長毎に区別して、複数の元部分データを生成し、この複数の元部分データの各々に対応して、前記データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成して、所望の分割数の分割データを生成するとともに、新たに発生させた乱数から処理単位ビット長の複数の乱数部分データを生成し、前記各分割部分データと該乱数部分データの排他的論理和により処理単位ビット長毎に再分割部分データを生成して、前記所望の分割数の再分割データを生成するデータ分割方法であり、ブロック暗号を構成要素に持つ擬似乱数生成関数を用いて第1の乱数を生成する手段と、前記データと前記第1の乱数から、前記秘密分散法を用いて複数の分割データを生成する手段と、前記複数の分割データを暗号データとして所定の記憶部に記憶する手段と、所定の時期に、前記記憶部に記憶された暗号データの生成時に用いられた擬似乱数生成関数よりも計算量大きい別の擬似乱数生成関数を用いて、第2の乱数を生成する手段と、前記記憶部に記憶された暗号データと前記第2の乱数から、前記秘密分散法を用いて新たな分割データを生成する手段と、前記記憶部に記憶された暗号データに代えて、前記新たな分割データを暗号データとして前記記憶部に記憶する手段と、を有することを特徴とする。

【0016】

請求項6記載の本発明は、請求項5記載の発明において、前記記憶部に記憶された暗号データのうち、復元可能な所定の個数の分割データの組み合わせから、前記秘密分散法を

用いて、前記データを復号する手段を有することを特徴とする。

【0017】

請求項7記載の本発明は、請求項5又は6記載の発明において、前記秘密分散法は、前記各分割部分データの定義式における乱数部分データを、該乱数部分データと該各乱数部分データに対応する新たな乱数部分データとの排他的論理和演算に置換した各再分割部分データの定義式により、各再分割部分データを生成することを特徴とする。

【0018】

請求項8記載の本発明は、請求項5又は6記載の発明において、前記秘密分散法は、前記各分割部分データの定義式における乱数部分データを、該乱数部分データに対応する新たな乱数部分データに置換した各再分割部分データの定義式により、各再分割部分データを生成することを特徴とする。

【0019】

請求項9記載の本発明は、請求項1乃至8のいずれか1項に記載の発明において、前記所定の時期は、コンピュータの計算能力をもとに判断された、前記擬似乱数関数に用いられたブロック暗号の脆弱時期であることを特徴とする。

【0020】

請求項10記載の本発明は、データをバーナム暗号を用いて秘匿するデータ秘匿方法であって、ブロック暗号を構成要素に持つ擬似乱数生成関数を用いて第1の乱数を生成するステップと、前記データと、前記第1の乱数との排他的論理和により暗号データを生成するステップと、生成された暗号データを所定の記憶部に記憶するステップと、所定の時期に、前記記憶部に記憶された暗号データの生成時に用いられた擬似乱数生成関数よりも計算量大きい別の擬似乱数生成関数を用いて、第2の乱数を生成するステップと、前記記憶部に記憶された暗号データと、前記第2の乱数との排他的論理和により暗号データを生成する暗号データ暗号化ステップと、前記記憶部に記憶された暗号データに代えて、前記暗号データ暗号化ステップで生成された暗号データを前記記憶部に記憶するステップと、を有することを特徴とする。

【0021】

請求項11記載の本発明は、データを秘密分散法を用いて秘匿するデータ秘匿方法であって、前記秘密分散法は、前記データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割方法であり、前記データを処理単位ビット長毎に区分けして、複数の元部分データを生成し、この複数の元部分データの各々に対応して、前記データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成して、所望の分割数の分割データを生成するとともに、新たに発生させた乱数から処理単位ビット長の複数の乱数部分データを生成し、前記各分割部分データと該乱数部分データの排他的論理和により処理単位ビット長毎に再分割部分データを生成して、前記所望の分割数の再分割データを生成するデータ分割方法であり、ブロック暗号を構成要素に持つ擬似乱数生成関数を用いて第1の乱数を生成するステップと、前記データと前記第1の乱数から、前記秘密分散法を用いて複数の分割データを生成するステップと、前記複数の分割データを暗号データとして所定の記憶部に記憶するステップと、所定の時期に、前記記憶部に記憶された暗号データの生成時に用いられた擬似乱数生成関数よりも計算量大きい別の擬似乱数生成関数を用いて、第2の乱数を生成するステップと、前記記憶部に記憶された暗号データと前記第2の乱数から、前記秘密分散法を用いて新たな分割データを生成するステップと、前記記憶部に記憶された暗号データに代えて、前記新たな分割データを暗号データとして前記記憶部に記憶するステップと、を有することを特徴とする。

【0022】

請求項12記載の本発明は、データをバーナム暗号を用いて秘匿するためのコンピュータが読み取り可能なデータ秘匿プログラムであって、ブロック暗号を構成要素に持つ擬似乱数生成関数を用いて第1の乱数を生成するステップと、前記データと、前記第1の乱数

との排他的論理和により暗号データを生成するステップと、生成された暗号データを所定の記憶部に記憶するステップと、所定の時期に、前記記憶部に記憶された暗号データの生成時に用いられた擬似乱数生成関数よりも計算量大きい別の擬似乱数生成関数を用いて、第2の乱数を生成するステップと、前記記憶部に記憶された暗号データと、前記第2の乱数との排他的論理和により暗号データを生成する暗号データ暗号化ステップと、前記記憶部に記憶された暗号データに代えて、前記暗号データ暗号化ステップで生成された暗号データを前記記憶部に記憶するステップと、を前記コンピュータに実行させることを特徴とする。

【0023】

請求項13記載の本発明は、請求項12記載の発明において、前記暗号データ暗号化ステップは、さらに、生成された暗号データと、前記記憶部に記憶された暗号データの生成時に用いられた乱数との排他的論理和により、暗号データを生成することを特徴とする。

【0024】

請求項14記載の本発明は、請求項12記載の発明において、前記記憶部に記憶された暗号データと、前記データから前記記憶部に記憶された暗号データを生成するまでに用いた乱数すべてとを結合した排他的論理和により、前記データを復号するステップを前記コンピュータに実行させることを特徴とする。

【0025】

請求項15記載の本発明は、請求項13記載の発明において、前記記憶部に記憶された暗号データと、前記記憶部に記憶された暗号データを生成するときに用いた乱数との排他的論理和により、前記データを復号するステップを有することを特徴とする。

【0026】

請求項16記載の本発明は、データを秘密分散法を用いて秘匿するためのコンピュータが読み取り可能なデータ秘匿プログラムであって、前記秘密分散法は、前記データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割方法であり、前記データを処理単位ビット長毎に区分けして、複数の元部分データを生成し、この複数の元部分データの各々に対応して、前記データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成して、所望の分割数の分割データを生成するとともに、新たに発生させた乱数から処理単位ビット長の複数の乱数部分データを生成し、前記各分割部分データと該乱数部分データの排他的論理和により処理単位ビット長毎に再分割部分データを生成して、前記所望の分割数の再分割データを生成するデータ分割方法であり、ブロック暗号を構成要素に持つ擬似乱数生成関数を用いて第1の乱数を生成するステップと、前記データと前記第1の乱数から、前記秘密分散法を用いて複数の分割データに分割するステップと、前記複数の分割データを暗号データとして所定の記憶部に記憶するステップと、所定の時期に、前記記憶部に記憶された暗号データの生成時に用いられた擬似乱数生成関数よりも計算量大きい別の擬似乱数生成関数を用いて、第2の乱数を生成するステップと、前記記憶部に記憶された暗号データと前記第2の乱数から、前記秘密分散法を用いて新たな分割データを生成するステップと、前記記憶部に記憶された暗号データに代えて、前記新たな分割データを暗号データとして前記記憶部に記憶するステップと、を前記コンピュータに実行させることを特徴とする。

【0027】

請求項17記載の本発明は、請求項16記載の発明において、前記記憶部に記憶された暗号データのうち、復元可能な所定の個数の分割データの組み合わせから、前記秘密分散法を用いて、前記データを復号するステップを前記コンピュータに実行させることを特徴とする。

【0028】

請求項18記載の本発明は、請求項16又は17記載の発明において、前記秘密分散法は、前記各分割部分データの定義式における乱数部分データを、該乱数部分データと該各

乱数部分データに対応する新たな乱数部分データとの排他的論理和演算に置換した各再分割部分データの定義式により、各再分割部分データを生成することを特徴とする。

【0029】

請求項19記載の本発明は、請求項16又は17記載の発明において、前記秘密分散法は、前記各分割部分データの定義式における乱数部分データを、該乱数部分データに対応する新たな乱数部分データに置換した各再分割部分データの定義式により、各再分割部分データを生成することを特徴とする。

【0030】

請求項20記載の本発明は、請求項12乃至19のいずれか1項に記載の発明において、前記所定の時期は、コンピュータの計算能力をもとに判断された、前記擬似乱数関数に用いられたブロック暗号の脆弱時期であることを特徴とする。

【発明の効果】

【0031】

本発明によれば、ブロック暗号を構成要素に持つ擬似乱数生成関数により生成された乱数を用いて、バーナム暗号により暗号化されたデータを、所定の時期にさらに、より計算量の大きい擬似乱数生成関数により生成された乱数を用いて、バーナム暗号によりさらに暗号化するので、長期間にわたってデータを秘匿し続けることができる。

【0032】

また、本発明によれば、ブロック暗号を用いた擬似乱数生成関数により生成された乱数を用いて、秘密分散法により生成された分割データを、所定の時期にさらに、秘密分散法により、より計算量の大きい擬似乱数生成関数により生成された乱数を用いて、さらに再分割するので、長期間にわたってデータを秘匿し続けることができる。

【0033】

特に、本発明における秘密分散法は、データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割方法であり、データを処理単位ビット長毎に区分けして、複数の元部分データを生成し、この複数の元部分データの各々に対応して、データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成して、所望の分割数の分割データを生成するとともに、新たに発生させた乱数から処理単位ビット長の複数の乱数部分データを生成し、各分割部分データと該乱数部分データの排他的論理和により処理単位ビット長毎に再分割部分データを生成して、所望の分割数の再分割データを生成するので、データを復元することなく、データを再分割することができるので、データをよりセキュアに管理することができる。

【発明を実施するための最良の形態】

【0034】

以下、本発明の実施の形態を図面を用いて説明する。

【0035】

<第1の実施の形態>

図1は、本発明の第1の実施の形態に係るデータ秘匿装置1の概略構成を示すブロック図である。図1に示すデータ秘匿装置1は、バーナム暗号を用いてデータを秘匿する装置であり、記憶部11、乱数生成部12、データ暗号部13、及びデータ復号部14を備えている。

【0036】

詳しくは、記憶部11は、秘匿したいデータS、データSからバーナム暗号により生成される暗号データI ($I = A, B, C, \dots$)、バーナム暗号に用いられるシードJ ($J = \alpha, \beta, \gamma, \dots$)などを記憶するものである。

【0037】

乱数生成部12は、ブロック暗号を構成要素に持つ擬似乱数アルゴリズムK ($K = F, G, H, \dots$)を用いてシードJから乱数を生成するようになっている。尚、擬似乱数アル

ゴリズムKを用いてシードJから生成された乱数を $K(J)$ と表す。

【0038】

データ暗号部13は、バーナム暗号を用いてデータを暗号化するもので、具体的には、データSと、乱数生成部12で生成された乱数 $K(J)$ との排他的論理和(XOR)を計算して、暗号データIを生成するようになっている。また、既にデータSが暗号化されているときは、暗号データIをさらに暗号化するようになり、暗号データIと、乱数生成部12で生成された乱数 $K(J)$ との排他的論理和(XOR)を計算して、暗号データI'を生成するようになっている。

【0039】

データ復号部14は、暗号データIをデータSに復号化するもので、バーナム暗号の性質から、暗号生成に使用した乱数 $K(J)$ と、暗号データIとの排他的論理和(XOR)を計算することで、データSを復号できるようになっている。

【0040】

ここで、上述したデータ秘匿装置1は、少なくとも演算機能および制御機能を備えた中央演算装置(CPU)、プログラムやデータを格納する機能を有するRAM等からなる主記憶装置(メモリ)を有する電子的な装置から構成されているものである。また、上記装置は、主記憶装置の他、ハードディスクなどの補助記憶装置を具備していてもよい。

【0041】

このうち、乱数生成部12、データ暗号部13、及びデータ復号部14は、上記CPUによる演算制御機能を具体的に示したものに他ならない。また、記憶部11は、上記主記憶装置及び補助記憶装置の機能を備えたものである。

【0042】

また、本実施の形態に係る各種処理を実行するプログラムは、前述した主記憶装置またはハードディスクに格納されているものである。そして、このプログラムは、ハードディスク、フレキシブルディスク、CD-ROM、MO、DVD-ROMなどのコンピュータ読み取り可能な記録媒体に記録することも、通信ネットワークを介して配信することも可能である。

【0043】

次に、図2を用いて、本実施の形態に係るデータ秘匿装置1の動作を説明する。

【0044】

まず、乱数生成部12が、記憶部11に記憶されたシード α をもとに擬似乱数生成アルゴリズムFにより、データSと同じ長さの乱数 $F(\alpha)$ を生成する(ステップS10)。

【0045】

次に、データSと乱数 $F(\alpha)$ から、バーナム暗号により、暗号データAを生成し、記憶部11に記憶する(ステップS20, S30)。即ち、 $A=S * F(\alpha)$ である。

【0046】

尚、擬似乱数生成アルゴリズムFは、その時点(データSを擬似乱数アルゴリズムFを用いて暗号化する時点)において、コンピュータの計算能力からブロック暗号が解読不能であると判断される程度の計算量を要するアルゴリズムで、暗号データ生成時にはデータ秘匿装置1に与えられている。

【0047】

時間の経過とともにコンピュータの性能が向上し、擬似乱数アルゴリズムFを構成するブロック暗号の脆弱化が予測された場合(例えば、具体的には、ブロック暗号解読に必要な計算量と、その時点で入手可能なコンピュータの計算能力の双方を鑑みて、数年程度で解読可能な場合など)には、解読に必要な計算量がより大きいブロック暗号を構成要素に持つ擬似乱数アルゴリズムGを用いて、記憶部11に記憶されたシード β をもとに、データSと同じ長さの乱数 $G(\beta)$ を生成する(ステップS40, S50)。

【0048】

次に、記憶部11に記憶された暗号データAと乱数 $G(\beta)$ とから、バーナム暗号を用いて、暗号データBを生成し、暗号データAに代えて記憶部11に記憶する(ステップS

60, S70)。即ち、 $B = A * G(\beta)$ である。

【0049】

尚、擬似乱数生成アルゴリズムGは、上述したように擬似乱数生成アルゴリズムFより計算量が必要とされ、その時点（暗号データAを擬似乱数アルゴリズムGを用いて暗号化する時点）において、コンピュータの計算能力からブロック暗号が解読不能であると判断される程度の計算量を要するアルゴリズムで、暗号データ生成時にはデータ秘匿装置1に与えられている。例えば、具体的には、擬似乱数生成アルゴリズムFの構成要素には128ビット鍵のAES（Advanced Encryption Standard）が用いられていた場合には、擬似乱数生成アルゴリズムGの構成要素としては、256ビット鍵のAESを用いるものである。

【0050】

以降は、上記ステップS40～S70の繰り返しである。即ち、時間の経過とともにコンピュータの性能が向上し、擬似乱数アルゴリズムGを構成するブロック暗号の脆弱化が予測された場合、解読に必要な計算量がより大きいブロック暗号を構成要素に持つ擬似乱数アルゴリズムHを用いて、記憶部11に記憶されたシード γ をもとに、データSと同じ長さの乱数H(γ)を生成し、記憶部11に記憶された暗号データBと乱数H(γ)とから、バーナム暗号を用いて、暗号データC($= B * H(\gamma)$)を生成し、暗号データBに代えて記憶部11に記憶する（ステップS40, S50, S60, S70）。

【0051】

次に、データSを復号する動作について説明する。記憶部11に記憶された暗号データからデータSを復号するには、データSから現時点の暗号データを生成するまでに用いた乱数すべてを取得し、該乱数すべてと、現時点の暗号データとの排他的論理和(XOR)を計算する（ステップS80, S90）。例えば、現時点において記憶部11に暗号データAが記憶されている場合には、用いた乱数はF(α)だけであるから、

$$\begin{aligned} A * F(\alpha) &= (S * F(\alpha)) * F(\alpha) \\ &= S * (F(\alpha) * F(\alpha)) \\ &= S * 0 \\ &= S \end{aligned}$$

となり、データSを得ることができる。

【0052】

また、現時点において記憶部11に暗号データBが記憶されている場合には、用いた乱数はF(α)及びG(β)であるから、

$$\begin{aligned} B * F(\alpha) * G(\beta) &= (A * G(\beta)) * F(\alpha) * G(\beta) \\ &= (S * F(\alpha) * G(\beta)) * F(\alpha) * G(\beta) \\ &= S * (F(\alpha) * F(\alpha)) * (G(\beta) * G(\beta)) \\ &= S * 0 * 0 \\ &= S \end{aligned}$$

となり、データSを得ることができる。

【0053】

また、現時点において記憶部11に暗号データCが記憶されている場合には、用いた乱数はF(α)、G(β)及びH(γ)であるから、

$$\begin{aligned} C * F(\alpha) * G(\beta) * H(\gamma) &= B * H(\gamma) * F(\alpha) * G(\beta) * H(\gamma) \\ &= (B * H(\gamma)) * F(\alpha) * G(\beta) * H(\gamma) \\ &= (S * F(\alpha) * G(\beta) * H(\gamma)) * F(\alpha) * G(\beta) * H(\gamma) \\ &= S * (F(\alpha) * F(\alpha)) * (G(\beta) * G(\beta)) * (H(\gamma) * H(\gamma)) \\ &= S * 0 * 0 * 0 \\ &= S \end{aligned}$$

となり、データSを得ることができる。

【0054】

従って、本実施の形態に係るデータ秘匿装置1によれば、ブロック暗号を構成要素に持つ擬似乱数生成関数により生成された乱数を用いて、バーナム暗号により暗号化されたデータを、所定の時期にさらに、より計算量の大きい擬似乱数生成関数により生成された乱数を用いて、バーナム暗号によりさらに暗号化するので、長期間にわたってデータを秘匿し続けることができる。

【0055】

尚、第1の実施の形態の変形として、データ暗号部13及びデータ復号部14の代わりに、以下に示すような暗号計算を行うデータ暗号部13'及びデータ復号部14'を備えたデータ秘匿装置1'（図示せず）を用いて本発明を実施してもよい。

【0056】

データ暗号部13'は、今回生成した乱数及び前回生成した乱数の双方を用いて計算するもので、例えば、上述した暗号データB及び暗号データCは、それぞれ
 $B = A * G(\beta) * F(\alpha)$ 、 $C = B * H(\gamma) * G(\beta)$ と表される。

【0057】

$$\begin{aligned} B &= A * G(\beta) * F(\alpha) \\ &= (S * F(\alpha)) * G(\beta) * F(\alpha) \\ &= S * (F(\alpha) * F(\alpha)) * G(\beta) \\ &= S * 0 * G(\beta) \\ &= S * G(\beta) \\ C &= B * H(\gamma) * G(\beta) \\ &= (S * G(\beta)) * H(\gamma) * G(\beta) \\ &= S * (G(\beta) * G(\beta)) * H(\gamma) \\ &= S * H(\gamma) \end{aligned}$$

従って、この方法では、脆弱化した擬似乱数アルゴリズムによる乱数成分を除くことができる。

【0058】

また、データ復号部14'は、今回生成した乱数だけを用いて排他的論理和(XOR)の計算をして、データを復号するもので、例えば、暗号データB及び暗号データCからデータSに復号するには、それぞれ、 $B * G(\beta)$ 、 $C * H(\gamma)$ により求められる。

【0059】

$$\begin{aligned} B * G(\beta) &= (S * G(\beta)) * G(\beta) \\ &= S * (G(\beta) * G(\beta)) \\ &= S * 0 \\ &= S \\ C * H(\gamma) &= (S * H(\gamma)) * H(\gamma) \\ &= S * (H(\gamma) * H(\gamma)) \\ &= S * 0 \\ &= S \end{aligned}$$

<第2の実施の形態>

(1. システム構成)

図3は、本発明の第2の実施の形態に係るデータ秘匿装置2の概略構成を示すブロック図である。図3に示すように、データ秘匿装置2は、通信ネットワーク4を介してハードウェア的に互いに独立した複数（本実施の形態では2とする）のデータ保管用サーバコンピュータ（以下、単に保管サーバとよぶ）3a、3bと接続されており、後述する独自の秘密分散アルゴリズムによる秘密分散法（以下、秘密分散法Aとよぶ）を用いて秘匿したいデータSを複数のデータに分割し、該分割データを保管サーバ3a、3bおよびデータ秘匿装置2に、保管するようになっている（本実施の形態では、秘匿したいデータSを3つの分割データD(1)、D(2)、D(3)に分割し、それぞれを保管サーバ3a、3bおよびデータ秘匿装置2に保管する）。尚、後述する秘密分散法Aは、バーナム暗号によりデータを秘匿するのではないが、ブロック暗号を用いた擬似乱数生成アルゴリズムにより生成さ

れた乱数列と、秘匿したいデータとの排他的論理和を計算してデータを暗号化をする暗号化方式であり、かつ、データを分割して秘匿するので、第1の実施の形態よりも、よりセキュアな方法である。

【0060】

尚、本実施の形態においては、データSを3分割して保管する場合を例に説明するが、本発明はデータSを3分割する場合に限定されるわけではなく、n分割（ $n = 2$ 以上の整数）の場合にも適用されるものである。また、本実施の形態においては、分割データD(1)、D(2)を保管サーバ3、分割データD(3)をデータ秘匿装置2に割り当てたが、どの分割データをどの装置に割り当ててもよいものである。さらには、データ分散という観点から、分割データを物理的に異なる装置に割り当てるようにしたが、分割されたデータそれぞれをデータ秘匿装置2に保管するようにしてもよいものである。

【0061】

詳しくは、記憶部21は、秘匿したいデータS、データSから秘密分散法Aにより生成される分割データの一つである分割データD(3)、秘密分散法Aに用いられるシードJ（ $J = \alpha, \beta, \gamma, \dots$ ）などを記憶するものである。

【0062】

乱数生成部22は、ブロック暗号を構成要素に持つ擬似乱数アルゴリズムK（ $K = F, G, H, \dots$ ）を用いてシードJから乱数を生成するようになっている。尚、擬似乱数アルゴリズムKを用いてシードJから生成された乱数を $K(J)$ と表す。

【0063】

分割データ生成部23は、データS及び乱数 $K(J)$ から秘密分散法Aを用いて複数の分割データD(1)、D(2)、D(3)を生成するようになっている。

【0064】

再分割データ生成部24は、既にデータSが分割されて、分割データD(1)、D(2)、D(3)が保持されているときは、さらに、この分割データD(1)、D(2)、D(3)及び乱数 $K(J)$ から秘密分散法Aを用いて複数の再分割データD'(1)、D'(2)、D'(3)を生成するようになっている。

【0065】

元データ復元部25は、複数の分割データD(1)、D(2)、D(3)（再分割データD'(1)、D'(2)、D'(3)）から秘密分散法Aを用いてデータSを復元するようになっている。

【0066】

データ通信部26は、保管サーバ3a及び3bとそれぞれ分割データの送受信を行うようになっている。

【0067】

ここで、上述したデータ秘匿装置2、保管サーバ3a、3bは、それぞれ少なくとも演算機能および制御機能を備えた中央演算装置（CPU）、プログラムやデータを格納する機能を有するRAM等からなる主記憶装置（メモリ）を有する電子的な装置から構成されているものである。また、上記装置は、主記憶装置の他、ハードディスクなどの補助記憶装置を具備していてもよい。

【0068】

このうち、データ秘匿装置2の乱数生成部22、分割データ生成部23、再分割データ生成部24、元データ復元部25、及び通信部26は、上記CPUによる演算制御機能を具体的に示したものに他ならない。また、記憶部21は、上記主記憶装置及び補助記憶装置の機能を備えたものである。

【0069】

また、本実施の形態に係る各種処理を実行するプログラムは、前述した主記憶装置またはハードディスクに格納されているものである。そして、このプログラムは、ハードディスク、フレキシブルディスク、CD-ROM、MO、DVD-ROMなどのコンピュータ読み取り可能な記録媒体に記録することも、通信ネットワークを介して配信することも可能である。

【0070】

(2. 秘密分散法 A)

ここで、本実施の形態における独自の秘密分散アルゴリズムによる秘密分散法 A について説明する。

【0071】

本実施形態における元データ（データ S に相当する）の分割および復元では、元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するが、この場合の処理単位ビット長は任意の値に設定することができ、元データを処理単位ビット長毎に区別して、この元部分データから分割部分データを分割数より 1 少ない数ずつ生成するので、元データのビット長が処理単位ビット長の（分割数-1）倍の整数倍に一致しない場合は、元データの末尾の部分に 0 を埋めるなどして元データのビット長を処理単位ビット長の（分割数-1）倍の整数倍に合わせることで本実施形態を適用することができる。

【0072】

また、上述した乱数も（分割数-1）個の元部分データの各々に対応して処理単位ビット長のビット長を有する（分割数-1）個の乱数部分データとして乱数生成部 22 から生成される。すなわち、乱数は処理単位ビット長毎に区別されて、処理単位ビット長のビット長を有する（分割数-1）個の乱数部分データとして生成される。更に、元データは処理単位ビット長に基づいて所望の分割数の分割データに分割されるが、この分割データの各々も（分割数-1）個の元部分データの各々に対応して処理単位ビット長のビット長を有する（分割数-1）個の分割部分データとして生成される。すなわち、分割データの各々は、処理単位ビット長毎に区別されて、処理単位ビット長のビット長を有する（分割数-1）個の分割部分データとして生成される。

【0073】

なお、以下の説明では、上述した元データ、乱数、分割データ、分割数および処理単位ビット長をそれぞれ S, R, D, n および b で表すとともに、また複数のデータや乱数などのうちの 1 つを表す変数として i ($=1 \sim n$) および j ($=1 \sim n-1$) を用い、（分割数-1）個の元部分データ、（分割数-1）個の乱数部分データ、および分割数 n 個の分割データ D のそれぞれのうちの 1 つをそれぞれ S (j), R (j) および D (i) で表記し、更に各分割データ D (i) を構成する複数 (n-1) の分割部分データを D (i, j) で表記するものとする。すなわち、S (j) は、元データ S の先頭から処理単位ビット長毎に区別して 1 番から順に採番した時の j 番目の元部分データを表すものである。

【0074】

この表記を用いると、元データ、乱数データ、分割データとこれらをそれぞれ構成する元部分データ、乱数部分データ、分割部分データは、次のように表記される。

【0075】

元データ S = (n-1) 個の元部分データ S (j)
 $= S(1), S(2), \dots, S(n-1)$
 乱数 R = (n-1) 個の乱数部分データ R (j)
 $= R(1), R(2), \dots, R(n-1)$
 n 個の分割データ D (i) = D (1), D (2), ..., D (n)
 各分割部分データ D (i, j)
 $= D(1, 1), D(1, 2), \dots, D(1, n-1)$
 $D(2, 1), D(2, 2), \dots, D(2, n-1)$
 $\dots \dots \dots$
 $D(n, 1), D(n, 2), \dots, D(n, n-1)$
 $(i=1 \sim n), (j=1 \sim n-1)$

本実施形態は、上述したように処理単位ビット長毎に区別される複数の部分データに対して元部分データと乱数部分データの排他的論理和演算 (XOR) を行って、詳しくは、元部分データと乱数部分データの排他的論理和演算 (XOR) からなる定義式を用いて、元データの分割を行うことを特徴とするものであり、上述したデータ分割処理に多項式や剰

余演算を用いる方法に比較して、コンピュータ処理に適したビット演算である排他的論理和 (XOR) 演算を用いることにより高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを生成することができるとともに、また分割データの保管に必要な記憶容量も分割数に比例した倍数の容量よりも小さくすることができる。更に、任意に定めた一定の長さ毎にデータの先頭から順に演算処理を行うストリーム処理により分割データが生成される。

【0076】

次に、フローチャートなどの図面も参照して、本実施の形態における秘密分散法Aの作用について説明するが、この説明の前に図4乃至8、図10および図12に示す記号の定義について説明する。

【0077】

(1) $\prod_{i=1}^n A(i)$ は、 $A(1)*A(2)*\dots*A(n)$ を意味するものとする。

【0078】

(2) $c(j, i, k)$ を $(n-1) \times (n-1)$ 行列である $U[n-1, n-1] \times (P[n-1, n-1])^{-(j-1)}$ の i 行 k 列の値と定義する。

【0079】

このとき $Q(j, i, k)$ を下記のように定義する。

【0080】

$c(j, i, k)=1$ のとき $Q(j, i, k)=R((n-1) \times m+k)$

$c(j, i, k)=0$ のとき $Q(j, i, k)=0$

ただし、 m は $m \geq 0$ の整数を表す。

【0081】

(3) $U[n, n]$ とは、 $n \times n$ 行列であって、 i 行 j 列の値を $u(i, j)$ で表すと、

$i+j \leq n+1$ のとき $u(i, j)=1$

$i+j > n+1$ のとき $u(i, j)=0$

である行列を意味するものとし、「上三角行列」ということとする。具体的には下記のような行列である。

【数1】

$$U[3, 3] = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad U[4, 4] = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

【0082】

(4) $P[n, n]$ とは、 $n \times n$ 行列であって、 i 行 j 列の値を $p(i, j)$ で表すと、

$j=i+1$ のとき $p(i, j)=1$

$i=1, j=n$ のとき $p(i, j)=1$

上記以外の場合 $p(i, j)=0$

である行列を意味するものとし、「回転行列」ということとする。具体的には下記のような行列であり、他の行列の右側からかけると当該他の行列の1列目を2列目へ、2列目を3列目へ、 \dots 、 $n-1$ 列目を n 列目へ、 n 列目を1列目へ移動させる作用がある。つまり、行列 P を他の行列に右側から複数回かけると、その回数分だけ各列を右方向へ回転させるように移動させることができる。

【数2】

$$P[3, 3] = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad U[4, 4] = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

【0083】

(5) A, Bを $n \times n$ 行列とすると、 $A \times B$ とは行列AとBの積を意味するものとする。行列の成分同士の計算規則は通常の数学で用いるものと同じである。

【0084】

(6) Aを $n \times n$ 行列とし、iを整数とすると、 A^i とは行列Aのi個の積を意味するものとする。また、 A^0 とは単位行列Eを意味するものとする。

【0085】

(7) 単位行列 $E[n, n]$ とは、 $n \times n$ 行列であって、i行j列の値を $e(i, j)$ で表すと、

i=j のとき $e(i, j)=1$

上記以外のとき $e(i, j)=0$

である行列を意味するものとする。具体的には下記のような行列である。Aを任意の $n \times n$ 行列とすると

$$A \times E = E \times A = A$$

となる性質がある。

【数3】

$$E[3, 3] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad E[4, 4] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

【0086】

次に、図4に示すフローチャートおよび図5および図6に示す具体的データなどを参照して、まず元データSの分割処理について説明する。これは、データ秘匿装置2の分割データ生成部23の機能を説明するものである。

【0087】

まず、元データSをデータ秘匿装置2に与える(図4のステップS201)。なお、本例では、元データSは、16ビットの「10110010 00110111」とする。

【0088】

次に、データ秘匿装置2は、分割数nとして3と指示する(ステップS203)。なお、この分割数 $n=3$ に従ってデータ秘匿装置2で生成される3個の分割データをD(1), D(2), D(3)とする。この分割データD(1), D(2), D(3)は、すべて元データのビット長と同じ16ビット長のデータである。

【0089】

それから、元データSを分割するために使用される処理単位ビット長bを8ビットと決定する(ステップS205)。この処理単位ビット長bは、利用者がデータ秘匿装置2に対して指定してもよいし、またはデータ秘匿装置2において予め定められた値を用いてもよい。なお、処理単位ビット長bは、任意のビット数でよいが、ここでは元データSを割り切ることができる8ビットとしている。従って、上記16ビットの「10110010 00110111

」の元データSは、8ビットの処理単位ビット長で分けられた場合の2個の元分割データS(1)およびS(2)は、それぞれ「10110010」および「00110111」となる。

【0090】

次のステップS207では、元データSのビット長が 8×2 の整数倍であるか否かを判定し、整数倍でない場合には、元データSの末尾を0で埋めて、 8×2 の整数倍に合わせる。なお、本例のように処理単位ビット長bが8ビットおよび分割数nが3に設定された場合における分割処理は、元データSのビット長として16ビットに限られるものでなく、処理単位ビット長 $b \times (\text{分割数}n-1) = 8 \times 2$ の整数倍の元データSに対して有効なものである。

【0091】

次に、ステップS209では、変数m、すなわち上述した整数倍を意味する変数mを0に設定する。本例のように、元データSが処理単位ビット長 $b \times (\text{分割数}n-1) = 8 \times 2 = 16$ ビットである場合には、変数mは0であるが、2倍の32ビットの場合には、変数mは1となり、3倍の48ビットの場合には、変数mは2となる。

【0092】

次に、元データSの $8 \times 2 \times m+1$ ビット目から 8×2 ビット分のデータが存在するか否かが判定される（ステップS211）。これは、このステップS211以降に示す分割処理を元データSの変数mで特定される処理単位ビット長 $b \times (\text{分割数}n-1) = 8 \times 2 = 16$ ビットに対して行った後、元データSとして次の16ビットがあるか否かを判定しているものである。本例のように元データSが16ビットである場合には、16ビットの元データSに対してステップS211以降の分割処理を1回行くと、後述するステップS219で変数mが+1されるが、本例の元データSでは変数mがm+1の場合に相当する17ビット以降のデータは存在しないので、ステップS211からステップS221に進むことになるが、今の場合は、変数mは0であるので、元データSの $8 \times 2 \times m+1$ ビット目は、 $8 \times 2 \times 0+1=1$ となり、元データSの16ビットの1ビット目から 8×2 ビット分にデータが存在するため、ステップS213に進む。

【0093】

ステップS213では、変数jを1から2(=分割数n-1)まで変えて、元データSの $8 \times (2 \times m+j-1)+1$ ビット目から8ビット分(=処理単位ビット長)のデータを元部分データS(2×m+j)に設定し、これにより元データSを処理単位ビット長で分けした2(分割数n-1)個の元部分データS(1),S(2)を次のように生成する。

【0094】

元データS=S(1),S(2)

第1の元部分データS(1)＝「10110010」

第2の元部分データS(2)＝「00110111」

次に、変数jを1から2(=分割数n-1)まで変えて、乱数部分データR(2×m+j)に乱数生成部22から発生する8ビットの長さの乱数を設定し、これにより乱数Rを処理単位ビット長で分けした2(分割数n-1)個の乱数部分データR(1),R(2)を次のように生成する（ステップS215）。

【0095】

乱数R=R(1),R(2)

第1の乱数部分データR(1)＝「10110001」

第2の乱数部分データR(2)＝「00110101」

次に、ステップS217において、変数iを1から3(=分割数n)まで変えるとともに、更に各変数iにおいて変数jを1から2(=分割数n-1)まで変えながら、ステップS217に示す分割データを生成するための元部分データと乱数部分データの排他的論理和からなる定義式により複数の分割データD(i)の各々を構成する各分割部分データD(i,2×m+j)を生成する。この結果、次に示すような分割データDが生成される。

【0096】

分割データD

＝3個の分割データD(i)=D(1),D(2),D(3)

第1の分割データD(1)

= 2個の分割部分データD(1, j)=D(1, 1), D(1, 2)
= 「00110110」, 「10110011」

第2の分割データD(2)

= 2個の分割部分データD(2, j)=D(2, 1), D(2, 2)
= 「00000011」, 「00000010」

第3の分割データD(3)

= 2個の分割部分データD(3, j)=D(3, 1), D(3, 2)
= 「10110001」, 「00110101」

なお、各分割部分データ(i, j)を生成するためのステップS 2 1 7に示す定義式は、本例のように分割数n=3の場合には、具体的には図6に示す表に記載されているものとなる。

図6に示す表から、分割部分データD(1, 1)を生成するための定義式は $S(1)*R(1)*R(2)$ であり、D(1, 2)の定義式は $S(2)*R(1)*R(2)$ であり、D(2, 1)の定義式は $S(1)*R(1)$ であり、D(2, 2)の定義式は $S(2)*R(2)$ であり、D(3, 1)の定義式は $R(1)$ であり、D(3, 2)の定義式は $R(2)$ である。また、図6に示す表には $m>0$ の場合の任意の整数についての一般的な定義式も記載されている。

【0097】

このように整数倍を意味する変数 $m=0$ の場合について分割データDを生成した後、次に変数 m を1増やし（ステップS 2 1 9）、ステップS 2 1 1に戻り、変数 $m+1$ に該当する元データSの17ビット以降について同様の分割処理を行おうとするが、本例の元データSは16ビットであり、17ビット以降のデータは存在しないので、ステップS 2 1 1からステップS 2 2 1に進み、上述したように生成した分割データD(1), D(2), D(3)を保管サーバ3及びデータ秘匿装置2にそれぞれ保存して、分割処理を終了する。なお、このように保管された分割データD(1), D(2), D(3)はそれぞれ単独では元データが推測できない。

【0098】

ここで、上述した図4のフローチャートのステップS 2 1 7における定義式による分割データの生成処理、具体的には分割数 $n=3$ の場合の分割データの生成処理について詳しく説明する。

【0099】

まず、整数倍を意味する変数 $m=0$ の場合には、ステップS 2 1 7に示す定義式から各分割データ $D(i)=D(1)\sim D(3)$ の各々を構成する各分割部分データ $D(i, 2\times m+j)=D(i, j)$ ($i=1\sim 3$, $j=1\sim 2$)は、次のようになる。

【0100】

$D(1, 1)=S(1)*Q(1, 1, 1)*Q(1, 1, 2)$
 $D(1, 2)=S(2)*Q(2, 1, 1)*Q(2, 1, 2)$
 $D(2, 1)=S(1)*Q(1, 2, 1)*Q(1, 2, 2)$
 $D(2, 2)=S(2)*Q(2, 2, 1)*Q(2, 2, 2)$
 $D(3, 1)=R(1)$
 $D(3, 2)=R(2)$

上記の6つの式のうち上から4つの式に含まれる $Q(j, i, k)$ を具体的に求める。

【0101】

これは $c(j, i, k)$ を 2×2 行列である $U[2, 2]\times (P[2, 2])^{(j-1)}$ の i 行 k 列の値としたとき下記のように定義される。

【0102】

$c(j, i, k)=1$ のとき $Q(j, i, k)=R(k)$
 $c(j, i, k)=0$ のとき $Q(j, i, k)=0$

ここで、

$j=1$ のときは

【数4】

$$\begin{aligned}
 U[2, 2] \times (P[2, 2])^{-(j-1)} &= U[2, 2] \times (P[2, 2])^{-0} \\
 &= U[2, 2] \times E[2, 2] \\
 &= U[2, 2] \\
 &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}
 \end{aligned}$$

【0103】

j=2のときは

【数5】

$$\begin{aligned}
 U[2, 2] \times (P[2, 2])^{-(j-1)} &= U[2, 2] \times (P[2, 2])^{-1} \\
 &= U[2, 2] \times P[2, 2] \\
 &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
 &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}
 \end{aligned}$$

【0104】

これを用いると、各分割部分データD(i, j)は次のような定義式により生成される。

【0105】

$$\begin{aligned}
 D(1, 1) &= S(1) * Q(1, 1, 1) * Q(1, 1, 2) = S(1) * R(1) * R(2) \\
 D(1, 2) &= S(2) * Q(2, 1, 1) * Q(2, 1, 2) = S(2) * R(1) * R(2) \\
 D(2, 1) &= S(1) * Q(1, 2, 1) * Q(1, 2, 2) = S(1) * R(1) * 0 = S(1) * R(1) \\
 D(2, 2) &= S(2) * Q(2, 2, 1) * Q(2, 2, 2) = S(2) * 0 * R(2) = S(2) * R(2)
 \end{aligned}$$

上述した各分割部分データD(i, j)を生成するための定義式は、図5にも図示されている。

【0106】

図5は、上述したように16ビットの元データSを8ビットの処理単位ビット長に基づいて分割数n=3で3分割する場合の各データと定義式および各分割部分データから元データを復元する場合の計算式などを示す表である。

【0107】

ここで、上述した定義式により分割データD(1), D(2), D(3)および各分割部分データD(1, 1), D(1, 2), D(2, 1), D(2, 2), D(3, 1), D(3, 2)を生成する過程と定義式の一般形について説明する。

【0108】

まず、第1の分割データD(1)に対しては、第1の分割部分データD(1, 1)は、上述した定義式S(1)*R(1)*R(2)で定義され、第2の分割部分データD(1, 2)は定義式S(2)*R(1)*R(2)で定義される。なお、この定義式の一般形は、D(1, j)に対してはS(j)*R(j)*R(j+1)であり、D(1, j+1)に対してS(j+1)*R(j)*R(j+1)である(jは奇数とする)。定義式に従って計算すると、D(1, 1)は00110110、D(1, 2)は10110011となるので、D(1)は00110110 10110011であ

る。なお、定義式の一般形は、図6にまとめて示されている。

【0109】

また、第2の分割データD(2)に対しては、D(2,1)はS(1)*R(1)で定義され、D(2,2)はS(2)*R(2)で定義される。この定義式の一般形は、D(2,j)に対してはS(j)*R(j)であり、D(2,j+1)に対してはS(j+1)*R(j+1)である(jは奇数とする)。定義式に従って計算すると、D(2,1)は00000011、D(2,2)は00000010となるので、D(2)は00000011 00000010である。

【0110】

更に第3の分割データD(3)に対しては、D(3,1)はR(1)で定義され、D(3,2)はR(2)で定義される。この定義式の一般形は、D(3,j)に対してはR(j)であり、D(3,j+1)に対してはR(j+1)である(jは奇数とする)。定義式に従って計算すると、D(3,1)は10110001、D(3,2)は00110101となるので、D(3)は10110001 00110101である。

【0111】

上記説明は、S,R,D(1),D(2),D(3)の長さを16ビットとしたが、データの先頭から上記分割処理を繰り返すことにより、どのような長さの元データSからでも分割データD(1),D(2),D(3)を生成することができる。また、処理単位ビット長bは任意にとることができ、元データSの先頭から順にb×2の長さ毎に上記分割処理を繰り返すことにより任意の長さの元データ、具体的には処理単位ビット長b×2の整数倍の長さの元データに対して適用することができる。なお、元データSの長さが処理単位ビット長b×2の整数倍でない場合は、例えば、データ末尾の部分を0で埋めるなどして元データSの長さを処理単位ビット長b×2の整数倍に合わせることで上記した本実施形態の分割処理を適用することができる。

【0112】

次に、図5の右側に示す表を参照して、分割データから元データを復元する処理について説明する。これは、データ秘匿装置2の元データ復元部25の機能を説明するものである。

【0113】

まず、データ秘匿装置2に元データSの復元を要求する。データ秘匿装置2は、自己および保管サーバ3から分割データD(1),D(2),D(3)を取得し、この取得した分割データD(1),D(2),D(3)から次に示すように元データSを復元する。

【0114】

まず、分割部分データD(2,1),D(3,1)から第1の元部分データS(1)を次のように生成することができる。

【0115】

$$\begin{aligned} D(2,1)*D(3,1) &= (S(1)*R(1))*R(1) \\ &= S(1)*(R(1)*R(1)) \\ &= S(1)*0 \\ &= S(1) \end{aligned}$$

具体的に計算すると、D(2,1)は00000011、D(3,1)は10110001なので、S(1)は10110010となる。

【0116】

また、別の分割部分データから次のように第2の元部分データS(2)を生成することができる。

【0117】

$$\begin{aligned} D(2,2)*D(3,2) &= (S(2)*R(2))*R(2) \\ &= S(2)*(R(2)*R(2)) \\ &= S(2)*0 \\ &= S(2) \end{aligned}$$

具体的に計算すると、D(2,2)は00000010、D(3,2)は00110101なので、S(2)は00110111となる。

【0118】

一般に、jを奇数として、

$$\begin{aligned} D(2, j) * D(3, j) &= (S(j) * R(j)) * R(j) \\ &= S(j) * (R(j) * R(j)) \\ &= S(j) * 0 \\ &= S(j) \end{aligned}$$

であるから、 $D(2, j) * D(3, j)$ を計算すれば、 $S(j)$ が求まる。

【0119】

また、一般に、jを奇数として、

$$\begin{aligned} D(2, j+1) * D(3, j+1) &= (S(j+1) * R(j+1)) * R(j+1) \\ &= S(j+1) * (R(j+1) * R(j+1)) \\ &= S(j+1) * 0 \\ &= S(j+1) \end{aligned}$$

であるから、 $D(2, j+1) * D(3, j+1)$ を計算すれば、 $S(j+1)$ が求まる。

【0120】

次に、 $D(1), D(3)$ を取得してSを復元する場合には、次のようになる。

【0121】

$$\begin{aligned} D(1, 1) * D(3, 1) * D(3, 2) &= (S(1) * R(1) * R(2)) * R(1) * R(2) = S(1) * (R(1) * R(1)) * (R(2) * R(2)) \\ &= S(1) * 0 * 0 \\ &= S(1) \end{aligned}$$

であるから、 $D(1, 1) * D(3, 1) * D(3, 2)$ を計算すれば、 $S(1)$ が求まる。具体的に計算すると、 $D(1, 1)$ は00110110、 $D(3, 1)$ は10110001、 $D(3, 2)$ は00110101なので、 $S(1)$ は10110010となる。

【0122】

また同様に、

$$\begin{aligned} D(1, 2) * D(3, 1) * D(3, 2) &= (S(2) * R(1) * R(2)) * R(1) * R(2) \\ &= S(2) * (R(1) * R(1)) * (R(2) * R(2)) \\ &= S(2) * 0 * 0 \\ &= S(2) \end{aligned}$$

であるから、 $D(1, 2) * D(3, 1) * D(3, 2)$ を計算すれば、 $S(2)$ が求まる。具体的に計算すると、 $D(1, 2)$ は10110011、 $D(3, 1)$ は10110001、 $D(3, 2)$ は00110101なので、 $S(2)$ は00110111となる。

【0123】

一般に、jを奇数として、

$$\begin{aligned} D(1, j) * D(3, j) * D(3, j+1) &= (S(j) * R(j) * R(j+1)) * R(j) * R(j+1) \\ &= S(j) * (R(j) * R(j)) * (R(j+1) * R(j+1)) \\ &= S(j) * 0 * 0 \\ &= S(j) \end{aligned}$$

であるから、 $D(1, j) * D(3, j) * D(3, j+1)$ を計算すれば、 $S(j)$ が求まる。

【0124】

また、一般に、jを奇数として、

$$\begin{aligned} D(1, j+1) * D(3, j) * D(3, j+1) &= (S(j+1) * R(j) * R(j+1)) * R(j) * R(j+1) \\ &= S(j+1) * (R(j) * R(j)) * (R(j+1) * R(j+1)) \\ &= S(j+1) * 0 * 0 \\ &= S(j+1) \end{aligned}$$

であるから、 $D(1, j+1) * D(3, j) * D(3, j+1)$ を計算すれば、 $S(j+1)$ が求まる。

【0125】

次に、 $D(1), D(2)$ を取得してSを復元する場合には、次のようになる。

【0126】

$$\begin{aligned} D(1, 1) * D(2, 1) &= (S(1) * R(1) * R(2)) * (S(1) * R(1)) \\ &= (S(1) * S(1)) * (R(1) * R(1)) * R(2) \end{aligned}$$

$$=0*0*R(2)$$

$$=R(2)$$

であるから、 $D(1,1)*D(2,1)$ を計算すれば、 $R(2)$ が求まる。具体的に計算すると、 $D(1,1)$ は00110110、 $D(2,1)$ は00000011なので、 $R(2)$ は00110101となる。

【0127】

また同様に、

$$D(1,2)*D(2,2)=(S(2)*R(1)*R(2))*(S(2)*R(2))$$

$$=(S(2)*S(2))*R(1)*(R(2)*R(2))$$

$$=0*R(1)*0$$

$$=R(1)$$

であるから、 $D(1,2)*D(2,2)$ を計算すれば、 $R(1)$ が求まる。具体的に計算すると、 $D(1,2)$ は10110011、 $D(2,2)$ は00000010なので、 $R(1)$ は10110001となる。

【0128】

この $R(1)$, $R(2)$ を使用して $S(1)$, $S(2)$ を求める。

【0129】

$$D(2,1)*R(1)=(S(1)*R(1))*R(1)$$

$$=S(1)*(R(1)*R(1))$$

$$=S(1)*0$$

$$=S(1)$$

であるから、 $D(2,1)*R(1)$ を計算すれば、 $S(1)$ が求まる。具体的に計算すると、 $D(2,1)$ は00000011、 $R(1)$ は10110001なので、 $S(1)$ は10110010となる。

【0130】

また同様に、

$$D(2,2)*R(2)=(S(2)*R(2))*R(2)$$

$$=S(2)*(R(2)*R(2))$$

$$=S(2)*0$$

$$=S(2)$$

であるから $D(2,2)*R(2)$ を計算すれば $S(2)$ が求まる。具体的に計算すると $D(2,2)$ は00000010、 $R(2)$ は00110101なので、 $S(2)$ は00110111となる。

【0131】

一般に、 j を奇数として、

$$D(1,j)*D(2,j)=(S(j)*R(j)*R(j+1))*(S(j)*R(j))$$

$$=(S(j)*S(j))*(R(j)*R(j))*R(j+1)$$

$$=0*0*R(j+1)$$

$$=R(j+1)$$

であるから $D(1,j)*D(2,j)$ を計算すれば $R(j+1)$ が求まる。

【0132】

また同様に、

$$D(1,j+1)*D(2,j+1)=(S(j+1)*R(j)*R(j+1))*(S(j+1)*R(j+1))$$

$$=(S(j+1)*S(j+1))*R(j)*(R(j+1)*R(j+1))$$

$$=0*R(j)*0$$

$$=R(j)$$

であるから $D(1,j+1)*D(2,j+1)$ を計算すれば $R(j)$ が求まる。

【0133】

この $R(j)$, $R(j+1)$ を使用して $S(j)$, $S(j+1)$ を求める。

【0134】

$$D(2,j)*R(j)=(S(j)*R(j))*R(j)$$

$$=S(j)*(R(j)*R(j))$$

$$=S(j)*0$$

$$=S(j)$$

であるから $D(2, j) \cdot R(j)$ を計算すれば $S(j)$ が求まる。

【0135】

また同様に、

$$\begin{aligned} D(2, j+1) \cdot R(j+1) &= (S(j+1) \cdot R(j+1)) \cdot R(j+1) \\ &= S(j+1) \cdot (R(j+1) \cdot R(j+1)) \\ &= S(j+1) \cdot 0 \\ &= S(j+1) \end{aligned}$$

であるから $D(2, j+1) \cdot R(j+1)$ を計算すれば $S(j+1)$ が求まる。

【0136】

上述したように、元データの先頭から処理単位ビット長 b に基づいて分割処理を繰り返して、分割データを生成した場合には、3つの分割データ $D(1), D(2), D(3)$ のすべてを用いなくても、3つの分割データのうち、2つの分割データを用いて上述したように元データを復元することができる。

【0137】

尚、本実施の形態に係るデータ秘匿装置2においては、3つの分割データ $D(1), D(2), D(3)$ を生成するようになっていたので、分割数が3の場合について説明したが、秘密分散法Aは、分割数が n の場合にも適用できるものである。

【0138】

次に、図7に示すフローチャートを参照して、分割数が n で、処理単位ビット長が b である場合の一般的な分割処理について説明する。

【0139】

まず、元データ S をデータ秘匿装置2に与える（ステップS401）。また、データ秘匿装置2に、分割数 n （ $n \geq 3$ である任意の整数）を指示する（ステップS403）。処理単位ビット長 b を決定する（ステップS405）。なお、 b は0より大きい任意の整数である。次に、元データ S のビット長が $b \times (n-1)$ の整数倍であるか否かを判定し、整数倍でない場合には、元データ S の末尾を0で埋める（ステップS407）。また、整数倍を意味する変数 m を0に設定する（ステップS409）。

【0140】

次に、元データ S の $b \times (n-1) \times m+1$ ビット目から $b \times (n-1)$ ビット分のデータが存在するか否かが判定される（ステップS411）。この判定の結果、データが存在しない場合は、ステップS421に進むことになるが、今の場合は、ステップS409で変数 m は0に設定された場合であるので、データが存在するため、ステップS413に進む。

【0141】

ステップS413では、変数 j を1から $n-1$ まで変えて、元データ S の $b \times ((n-1) \times m+j-1) + 1$ ビット目から b ビット分のデータを元部分データ $S((n-1) \times m+j)$ に設定する処理を繰り返して、これにより元データ S を処理単位ビット長 b で分けした $(n-1)$ 個の元部分データ $S(1), S(2), \dots, S(n-1)$ が生成される。

【0142】

次に、変数 j を1から $n-1$ まで変えて、乱数部分データ $R((n-1) \times m+j)$ に乱数生成部22から発生する処理単位ビット長 b の乱数を設定し、これにより乱数 R を処理単位ビット長 b で分けした $n-1$ 個の乱数部分データ $R(1), R(2), \dots, R(n-1)$ が生成される（ステップS415）。

【0143】

次に、ステップS417において、変数 i を1から n まで変えるとともに、更に各変数 i において変数 j を1から $n-1$ まで変えながら、ステップS417に示す分割データを生成するための定義式により複数の分割データ $D(i)$ の各々を構成する各分割部分データ $D(i, (n-1) \times m+j)$ を生成する。この結果、次に示すような分割データ D が生成される。

【0144】

分割データ D

$$= n \text{ 個の分割データ } D(i) = D(1), D(2), \dots, D(n)$$

=n-1個の分割部分データ $D(n, j)=D(n, 1), D(n, 2), \dots, D(n, n-1)$

$$D(1, j) * D(1, j+1) = (S(j) * R(j) * R(j+1)) * (S(j+1) * R(j+1))$$

$$\begin{aligned}
&=S(j)*S(j+1)*R(j)*((R(j+1)*R(j+1))) \\
&=S(j)*S(j+1)*R(j)*0 \\
&=S(j)*S(j+1)*R(j) \\
D(2,j)*D(2,j+1)&=(S(j)*R(j))*(S(j+1)*R(j)*R(j+1)) \\
&=S(j)*S(j+1)*(R(j)*R(j))*R(j+1) \\
&=S(j)*S(j+1)*0*R(j+1) \\
&=S(j)*S(j+1)*R(j+1) \\
D(3,j)*D(3,j+1)&=R(j)*R(j+1)
\end{aligned}$$

となるからである。

【0151】

また、この場合、3つの分割データのうち2つから、元データを復元することができるという特性は失われていない。これは、D(1)、D(2)を取得してSを復元する場合には、図8におけるD(1)、D(2)は、図6におけるD(1)、D(2)を構成する分割部分データを入れ替えたものにすぎないので、明らかにこれらから元データを復元することができ、また、D(1)とD(3)またはD(2)とD(3)を取得してSを復元する場合には、D(3)は乱数のみからなる分割データであるので、D(1)またはD(2)の分割部分データ毎に必要な個数の乱数との排他的論理和演算を行うことにより、乱数部分を消去して元データを復元することができるからである。

【0152】

次に、一旦分割された分割データにさらに乱数を与えて新たな分割データ（再分割データ）を生成する再分割処理について説明する。データ秘匿装置2の再分割データ生成部24の機能を説明するものであるが、これに関しても、分割数が3の場合を例に説明する。尚、本実施の形態における再分割処理は、2つの方法があるので、以下、それぞれについて説明する。

【0153】

（2-1．乱数追加注入方式）

図9は、乱数追加注入方式におけるデータ再分割処理の概要を説明するフローチャート図である。同図によれば、まず分割データD(1)、D(2)、D(3)を取得し（ステップS501）、次に、乱数生成部22で再分割の際に用いる乱数R'を生成する（ステップS503）。

【0154】

次に、分割データD(1)、D(2)、D(3)それぞれに乱数R'を所定のルールで注入する（ステップS505）。これは、後述するようなルールにより分割データD(1)、D(2)、D(3)の分割部分データと乱数R'の乱数部分データの排他的論理和をとり、新たな分割データD'(1)、D'(2)、D'(3)を生成するものである（ステップS507）。

【0155】

図10は、元データSを、元データの半分の長さの処理単位ビット長bに基づいて分割数n=3で3分割する場合の分割部分データの定義式、乱数の再注入後の分割部分データの定義式、および各分割部分データから元データを復元する場合の計算式などを示す表である。

【0156】

ここで、分割部分データD(i,j)の定義式について説明する。

【0157】

まず、第1の分割データD(1)に対しては、図8に示すように、第1の分割部分データD(1,1)は、定義式S(1)*R(1)*R(2)で定義され、第2の分割部分データD(1,2)は定義式S(2)*R(2)で定義される。なお、この定義式の一般形は、D(1,j)に対してはS(j)*R(j)*R(j+1)であり、D(1,j+1)に対してS(j+1)*R(j+1)である（jは奇数とする）。

【0158】

また、第2の分割データD(2)に対しては、図8に示すように、D(2,1)はS(1)*R(1)で定義され、D(2,2)はS(2)*R(1)*R(2)で定義される。この定義式の一般形は、D(2,j)に対して

は $S(j)*R(j)$ であり、 $D(2,j+1)$ に対しては $S(j+1)*R(j)*R(j+1)$ である（ j は奇数とする）。

【0159】

更に第3の分割データ $D(3)$ に対しては、図8に示すように、 $D(3,1)$ は $R(1)$ で定義され、 $D(3,2)$ は $R(2)$ で定義される。この定義式の一般形は、 $D(3,j)$ に対しては $R(j)$ であり、 $D(3,j+1)$ に対しては $R(j+1)$ である（ j は奇数とする）。

【0160】

次に、新たな乱数 R' 注入後の分割部分データ $D'(i,j)$ の定義式について説明する。

【0161】

まず、第1の分割データ $D'(1)$ に対しては、図10に示すように、第1の分割部分データ $D'(1,1)$ は、定義式 $D(1,1)*R'(1)*R'(2)$ 、即ち、 $S(1)*R(1)*R(2)*R'(1)*R'(2)$ で定義され、第2の分割部分データ $D'(1,2)$ は、定義式 $D(1,2)*R'(2)$ 、即ち、 $S(2)*R(2)*R'(2)$ で定義される。なお、この定義式の一般形は、 $D'(1,j)$ に対しては $D(1,j)*R'(j)*R'(j+1)$ であり、 $D'(1,j+1)$ に対しては $D(1,j+1)*R'(j+1)$ である（ j は奇数とする）。

【0162】

また、第2の分割データ $D'(2)$ に対しては、図10に示すように、 $D'(2,1)$ は $D(2,1)*R'(1)$ 、即ち、 $S(1)*R(1)*R'(1)$ で定義され、 $D'(2,2)$ は $D(2,2)*R'(1)*R'(2)$ 、即ち、 $S(2)*R(1)*R(2)*R'(1)*R'(2)$ で定義される。この定義式の一般形は、 $D'(2,j)$ に対しては $D(2,j)*R'(j)$ であり、 $D'(2,j+1)$ に対しては $D(2,j+1)*R'(j)*R'(j+1)$ である（ j は奇数とする）。

【0163】

また、第3の分割データ $D'(3)$ に対しては、図10に示すように、 $D'(3,1)$ は $D(3,1)*R'(1)$ 、即ち、 $R(1)*R'(1)$ で定義され、 $D'(3,2)$ は $D(3,2)*R'(2)$ 、即ち、 $R(2)*R'(2)$ で定義される。この定義式の一般形は、 $D'(3,j)$ に対しては $D(3,j)*R'(j)$ であり、 $D'(3,j+1)$ に対しては $D(3,j+1)*R'(j+1)$ である（ j は奇数とする）。

【0164】

このように、再分割部分データ $D'(i,j)$ はそれぞれ、分割部分データ $D(i,j)$ に、分割部分データ $D(i,j)$ の定義式で注入されていた乱数部分データ $R(j)$ に対応する乱数部分データ $R'(j)$ を注入して排他的論理和を計算して求めるものである。

【0165】

次に、図10の右側に示す表を参照して、再分割データから元データを復元する処理について説明する。これは、データ秘匿装置2の元データ復元部25の機能を説明するものである。

【0166】

まず、分割部分データ $D'(2,1)$ 、 $D'(3,1)$ から第1の元部分データ $S(1)$ を次のように生成することができる。

【0167】

$$\begin{aligned} D'(2,1)*D'(3,1) &= (S(1)*R(1)*R'(1))*(R(1)*R'(1)) \\ &= S(1)*(R(1)*R(1))* (R'(1)*R'(1)) \\ &= S(1)*0*0 \\ &= S(1) \end{aligned}$$

また、別の分割部分データから次のように第2の元部分データ $S(2)$ を生成することができる。

【0168】

$$\begin{aligned} D'(2,2)*D'(3,1)*D'(3,2) &= (S(2)*R(1)*R(2)*R'(1)*R'(2))* \\ &\quad (R(1)*R'(1))*(R(2)*R'(2)) \\ &= S(2)*(R(1)*R(1))*(R(2)*R(2))* \\ &\quad (R'(1)*R'(1))*(R'(2)*R'(2)) \\ &= S(2)*0*0*0*0 \\ &= S(2) \end{aligned}$$

一般に、 j を奇数として、

$$\begin{aligned}
 D'(2, j) * D'(3, j) &= (S(j) * R(j) * R'(j)) * (R(j) * R'(j)) \\
 &= S(j) * (R(j) * R(j)) * (R'(j) * R'(j)) \\
 &= S(j) * 0 * 0 \\
 &= S(j)
 \end{aligned}$$

であるから、 $D'(2, j) * D'(3, j)$ を計算すれば、 $S(j)$ が求まる。

【0169】

また、一般に、 j を奇数として、

$$\begin{aligned}
 D'(2, j+1) * D'(3, j) * D'(3, j+1) &= (S(j+1) * R(j) * R(j+1) * R'(j) * R'(j+1)) * \\
 &\quad (R(j) * R'(j)) * (R(j+1) * R'(j+1)) \\
 &= S(j+1) * ((R(j) * R(j)) * (R(j+1) * R(j+1)) * \\
 &\quad * (R'(j) * R'(j)) * (R'(j+1) * R'(j+1))) \\
 &= S(j+1) * 0 * 0 * 0 \\
 &= S(j+1)
 \end{aligned}$$

であるから、 $D'(2, j+1) * D'(3, j) * D'(3, j+1)$ を計算すれば、 $S(j+1)$ が求まる。

【0170】

次に、 $D'(1), D'(3)$ を取得して S を復元する場合には、次のようになる。

【0171】

$$\begin{aligned}
 D'(1, 1) * D'(3, 1) * D'(3, 2) &= (S(1) * R(1) * R(2) * R'(1) * R'(2)) * \\
 &\quad (R(1) * R'(1)) * (R(2) * R'(2)) \\
 &= S(1) * (R(1) * R(1)) * (R(2) * R(2)) * \\
 &\quad (R'(1) * R'(1)) * (R'(2) * R'(2)) \\
 &= S(1) * 0 * 0 * 0 \\
 &= S(1)
 \end{aligned}$$

であるから、 $D'(1, 1) * D'(3, 1) * D'(3, 2)$ を計算すれば、 $S(1)$ が求まる。

【0172】

また同様に、

$$\begin{aligned}
 D'(1, 2) * D'(3, 2) &= (S(2) * R(2) * R'(2)) * (R(2) * R'(2)) \\
 &= S(2) * (R(2) * R(2)) * (R'(2) * R'(2)) \\
 &= S(2) * 0 * 0 \\
 &= S(2)
 \end{aligned}$$

であるから、 $D'(1, 2) * D'(3, 2)$ を計算すれば、 $S(2)$ が求まる。

【0173】

一般に、 j を奇数として、

$$\begin{aligned}
 D'(1, j) * D'(3, j) * D'(3, j+1) &= (S(j) * R(j) * R(j+1) * R'(j) * R'(j+1)) * \\
 &\quad (R(j) * R'(j)) * (R(j+1) * R'(j+1)) \\
 &= S(j) * (R(j) * R(j)) * (R(j+1) * R(j+1)) * \\
 &\quad (R'(j) * R'(j)) * (R'(j+1) * R'(j+1)) \\
 &= S(j) * 0 * 0 * 0 \\
 &= S(j)
 \end{aligned}$$

であるから、 $D'(1, j) * D'(3, j) * D'(3, j+1)$ を計算すれば、 $S(j)$ が求まる。

【0174】

また、一般に、 j を奇数として、

$$\begin{aligned}
 D'(1, j+1) * D'(3, j+1) &= (S(j+1) * R(j+1) * R'(j+1)) * (R(j+1) * R'(j+1)) \\
 &= S(j+1) * (R(j+1) * R(j+1)) * (R'(j+1) * R'(j+1)) \\
 &= S(j+1) * 0 * 0 \\
 &= S(j+1)
 \end{aligned}$$

であるから、 $D'(1, j+1) * D'(3, j+1)$ を計算すれば、 $S(j+1)$ が求まる。

【0175】

次に、 $D'(1), D'(2)$ を取得して S を復元する場合には、次のようになる。

【0176】

$$\begin{aligned}
D'(1,1)*D'(2,1) &= (S(1)*R(1)*R(2)*R'(1)*R'(2))*(S(1)*R(1)*R'(1)) \\
&= (S(1)*S(1))*(R(1)*R(1))*(R'(1)*R'(1))*R(2)*R'(2) \\
&= 0*0*R(2)*R'(2) \\
&= R(2)*R'(2)
\end{aligned}$$

であるから、 $D'(1,1)*D'(2,1)$ を計算すれば、 $R(2)*R'(2)$ が求まる。

【0177】

また同様に、

$$\begin{aligned}
D'(1,2)*D'(2,2) &= (S(2)*R(2)*R'(2))*(S(2)*R(1)*R(2)*R'(1)*R'(2)) \\
&= (S(2)*S(2))*R(1)*R'(1)*(R(2)*R(2))*(R'(2)*R'(2)) \\
&= 0*R(1)*R'(1)*0*0 \\
&= R(1)*R'(1)
\end{aligned}$$

であるから、 $D'(1,2)*D'(2,2)$ を計算すれば、 $R(1)*R'(1)$ が求まる。

【0178】

この $R(1)*R'(1)$ 、 $R(2)*R'(2)$ を使用して $S(1)$ 、 $S(2)$ を求める。

【0179】

$$\begin{aligned}
D'(2,1)*R(1)*R'(1) &= (S(1)*R(1)*R'(1))*R(1)*R'(1) \\
&= S(1)*(R(1)*R(1))*(R'(1)*R'(1)) \\
&= S(1)*0*0 \\
&= S(1)
\end{aligned}$$

であるから、 $D'(2,1)*R(1)*R'(1)$ を計算すれば、 $S(1)$ が求まる。

【0180】

また同様に、

$$\begin{aligned}
D'(1,2)*R(2)*R'(2) &= (S(2)*R(2)*R'(2))*R(2)*R'(2) \\
&= S(2)*(R(2)*R(2))*(R'(2)*R'(2)) \\
&= S(2)*0*0 \\
&= S(2)
\end{aligned}$$

であるから $D'(2,2)*R(2)*R'(2)$ を計算すれば $S(2)$ が求まる。

【0181】

一般に、 j を奇数として、

$$\begin{aligned}
D'(1,j)*D'(2,j) &= (S(j)*R(j)*R(j+1)*R'(j)*R'(j+1))*(S(j)*R(j)*R'(j)) \\
&= (S(j)*S(j))*(R(j)*R(j))*(R'(j)*R'(j))*R(j+1)*R'(j+1) \\
&= 0*0*R(j+1)*R'(j+1) \\
&= R(j+1)*R'(j+1)
\end{aligned}$$

であるから $D'(1,j)*D'(2,j)$ を計算すれば $R(j+1)*R'(j+1)$ が求まる。

【0182】

また同様に、

$$\begin{aligned}
D'(1,j+1)*D'(2,j+1) &= (S(j+1)*R(j+1)*R'(j+1))* \\
&\quad (S(j+1)*R(j)*R(j+1)*R'(j)*R'(j+1)) \\
&= (S(j+1)*S(j+1))*R(j)*R'(j)* \\
&\quad (R(j+1)*R(j+1))*(R'(j+1)*R'(j+1)) \\
&= 0*R(j)*R'(j)*0*0 \\
&= R(j)*R'(j)
\end{aligned}$$

であるから $D'(1,j+1)*D'(2,j+1)$ を計算すれば $R(j)*R'(j)$ が求まる。

【0183】

この $R(j)*R'(j)$ 、 $R(j+1)*R'(j+1)$ を使用して $S(j)$ 、 $S(j+1)$ を求める。

【0184】

$$\begin{aligned}
D'(2,j)*R(j)*R'(j) &= (S(j)*R(j)*R'(j))*R(j)*R'(j) \\
&= S(j)*(R(j)*R(j))*(R'(j)*R'(j)) \\
&= S(j)*0*0 \\
&= S(j)
\end{aligned}$$

であるから $D'(2, j) * R(j) * R'(j)$ を計算すれば $S(j)$ が求まる。

【0185】

また同様に、

$$\begin{aligned} D'(1, j+1) * R(j+1) * R'(j+1) &= (S(j+1) * R(j+1) * R'(j+1)) * R(j+1) * R'(j+1) \\ &= S(j+1) * (R(j+1) * R(j+1)) * (R'(j+1) * R'(j+1)) \\ &= S(j+1) * 0 * 0 \\ &= S(j+1) \end{aligned}$$

であるから $D'(1, j+1) * R(j+1) * R'(j+1)$ を計算すれば $S(j+1)$ が求まる。

【0186】

以上、乱数追加注入方式により再分割データを生成した場合には、3つの再分割データ $D'(1), D'(2), D'(3)$ のすべてを用いなくても、3つの再分割データのうち、2つの再分割データを用いて上述したように元データを復元することができる。

【0187】

また、乱数追加注入方式においては、一旦元データを復元することなく（元データが見える形で現れない）、データの再分割処理を行うことができるので、よりセキュアなデータ管理が可能となる。

【0188】

（2-2. 乱数書き換え方式）

図1 1は、乱数書き換え方式におけるデータ再分割処理の概要を説明するフローチャート図である。同図によれば、まず分割データ $D(1), D(2), D(3)$ を取得し（ステップ S601）、次に、乱数生成部22で再分割の際に用いる乱数 R' を生成する（ステップ S603）。

【0189】

次に、分割データ $D(1), D(2), D(3)$ それぞれに乱数 R' を上述した乱数追加注入方式により注入する（ステップ S605）。次に、乱数 R' を注入された分割データから旧乱数である R を消去して、新たな再分割データ $D'(1), D'(2), D'(3)$ を生成する（ステップ S607, S609）。

【0190】

図1 2は、元データ S を、元データの半分の長さの処理単位ビット長 b に基づいて分割数 $n=3$ で3分割する場合の分割部分データの定義式、乱数 R' の再注入後の分割部分データの定義式、さらに乱数 R を消去後の分割部分データの定義式および各分割部分データから元データを復元する場合の計算式などを示す表である。

【0191】

本方式においては、ステップ S605 までは、上述した乱数追加注入方式と同様であるため、説明は省略し、古い乱数 R を消去した分割部分データの定義式について説明する。

【0192】

まず、第1の分割データ $D'(1)$ に対しては、図1 2に示すように、第1の分割部分データ $D'(1, 1)$ は、定義式 $(S(1) * R(1) * R(2) * R'(1) * R'(2)) * (R(1) * R(2))$ 、即ち、 $S(1) * R'(1) * R'(2)$ で定義され、第2の分割部分データ $D'(1, 2)$ は、定義式 $(S(2) * R(2) * R'(2)) * R(2)$ 、即ち、 $S(2) * R'(2)$ で定義される。なお、この定義式の一般形は、 $D'(1, j)$ に対しては $S(j) * R'(j) * R'(j+1)$ であり、 $D'(1, j+1)$ に対して $S(j+1) * R'(j+1)$ である（ j は奇数とする）。

【0193】

また、第2の分割データ $D'(2)$ に対しては、図1 2に示すように、 $D'(2, 1)$ は $(S(1) * R(1) * R'(1)) * R(1)$ 、即ち、 $S(1) * R'(1)$ で定義され、 $D'(2, 2)$ は $(S(2) * R(1) * R(2) * R'(1) * R'(2)) * R(1) * R(2)$ 、即ち、 $S(2) * R'(1) * R'(2)$ で定義される。この定義式の一般形は、 $D'(2, j)$ に対しては $S(j) * R'(j) *$ であり、 $D(2, j+1)$ に対しては $S(j+1) * R'(j) * R'(j+1)$ である（ j は奇数とする）。

【0194】

また、第3の分割データ $D'(3)$ に対しては、図1 2に示すように、 $D'(3, 1)$ は $(R(1) * R$

' (1))*R(1)、即ち、R' (1)で定義され、D' (3,2)は(R(2)* R' (2))* R(2)、即ち、R' (2)で定義される。この定義式の一般形は、D' (3,j)に対してはR' (j)*であり、D(3,j+1)に対してはR' (j+1)である(jは奇数とする)。

【0195】

このように、再分割部分データD' (i,j)はそれぞれ、分割部分データD (i,j)に、分割部分データD (i,j)の定義式で注入されていた乱数部分データR(j)に対応する乱数部分データR' (j)を注入した後、さらに乱数部分データR(j)を消去するように乱数部分データR(j)を注入して排他的論理和を計算し、求めるものである。

【0196】

その結果、もとの分割部分データD (i,j)の定義式において、乱数部分データR(j)を乱数部分データR' (j)に置換したものが、再分割部分データD' (i,j)の定義式となる。

【0197】

次に、図12の右側に示す表を参照して、再分割データから元データを復元する処理について説明する。これは、データ秘匿装置2の元データ復元部25の機能を説明するものである。

【0198】

まず、分割部分データD' (2,1), D' (3,1)から第1の元部分データS(1)を次のように生成することができる。

【0199】

$$\begin{aligned} D' (2,1)*D' (3,1) &= (S(1)*R' (1))*R' (1) \\ &= S(1)*(R' (1)*R' (1)) \\ &= S(1)*0 \\ &= S(1) \end{aligned}$$

また、別の分割部分データから次のように第2の元部分データS(2)を生成することができる。

【0200】

$$\begin{aligned} D' (2,2)*D' (3,1)*D' (3,2) &= (S(2)*R' (1)*R' (2))*R' (1)*R' (2) \\ &= S(2)*(R' (1)*R' (1))*(R' (2)*R' (2)) \\ &= S(2)*0*0 \\ &= S(2) \end{aligned}$$

一般に、jを奇数として、

$$\begin{aligned} D' (2,j)*D' (3,j) &= (S(j)*R' (j))*R' (j) \\ &= S(j)*(R' (j)*R' (j)) \\ &= S(j)*0 \\ &= S(j) \end{aligned}$$

であるから、D' (2,j)*D' (3,j)を計算すれば、S(j)が求まる。

【0201】

また、一般に、jを奇数として、

$$\begin{aligned} D' (2,j+1)*D' (3,j)*D' (3,j+1) &= (S(j+1)*R' (j)*R' (j+1))*R' (j)*R' (j+1) \\ &= S(j+1)*(R' (j)*R' (j))*(R' (j+1)*R' (j+1)) \\ &= S(j+1)*0*0 \\ &= S(j+1) \end{aligned}$$

であるから、D' (2,j+1)*D' (3,j)*D' (3,j+1)を計算すれば、S(j+1)が求まる。

【0202】

次に、D' (1), D' (3)を取得してSを復元する場合には、次のようになる。

【0203】

$$\begin{aligned} D' (1,1)*D' (3,1)*D' (3,2) &= (S(1)*R' (1)*R' (2))*R' (1)*R' (2) \\ &= S(1)*(R' (1)*R' (1))*(R' (2)*R' (2)) \\ &= S(1)*0*0 \\ &= S(1) \end{aligned}$$

であるから、 $D'(1,1)*D'(3,1)*D'(3,2)$ を計算すれば、 $S(1)$ が求まる。

【0204】

また同様に、

$$\begin{aligned} D'(1,2)*D'(3,2) &= (S(2)*R'(2))*R'(2) \\ &= S(2)*(R'(2)*R'(2)) \\ &= S(2)*0 \\ &= S(2) \end{aligned}$$

であるから、 $D'(1,2)*D'(3,2)$ を計算すれば、 $S(2)$ が求まる。

【0205】

一般に、 j を奇数として、

$$\begin{aligned} D'(1,j)*D'(3,j)*D'(3,j+1) &= (S(j)*R'(j)*R'(j+1))*R'(j)*R'(j+1) \\ &= S(j)*(R'(j)*R'(j))*(R'(j+1)*R'(j+1)) \\ &= S(j)*0*0 \\ &= S(j) \end{aligned}$$

であるから、 $D'(1,j)*D'(3,j)*D'(3,j+1)$ を計算すれば、 $S(j)$ が求まる。

【0206】

また、一般に、 j を奇数として、

$$\begin{aligned} D'(1,j+1)*D'(3,j+1) &= (S(j+1)*R'(j+1))*R'(j+1) \\ &= S(j+1)*(R'(j+1)*R'(j+1)) \\ &= S(j+1)*0 \\ &= S(j+1) \end{aligned}$$

であるから、 $D'(1,j+1)*D'(3,j+1)$ を計算すれば、 $S(j+1)$ が求まる。

【0207】

次に、 $D'(1), D'(2)$ を取得して S を復元する場合には、次のようになる。

【0208】

$$\begin{aligned} D'(1,1)*D'(2,1) &= (S(1)*R'(1)*R'(2))*(S(1)*R'(1)) \\ &= (S(1)*S(1))*(R'(1)*R'(1))*R'(2) \\ &= 0*0*R'(2) \\ &= R'(2) \end{aligned}$$

であるから、 $D'(1,1)*D'(2,1)$ を計算すれば、 $R'(2)$ が求まる。

【0209】

また同様に、

$$\begin{aligned} D'(1,2)*D'(2,2) &= (S(2)*R'(2))*(S(2)*R'(1)*R'(2)) \\ &= (S(2)*S(2))*(R'(2)*R'(2))*R'(1) \\ &= 0*0*R'(1) \\ &= R'(1) \end{aligned}$$

であるから、 $D'(1,2)*D'(2,2)$ を計算すれば、 $R'(1)$ が求まる。

【0210】

この $R'(1), R'(2)$ を使用して $S(1), S(2)$ を求める。

【0211】

$$\begin{aligned} D'(2,1)*R'(1) &= (S(1)*R'(1))*R'(1) \\ &= S(1)*(R'(1)*R'(1)) \\ &= S(1)*0 \\ &= S(1) \end{aligned}$$

であるから、 $D'(2,1)*R'(1)$ を計算すれば、 $S(1)$ が求まる。

【0212】

また同様に、

$$\begin{aligned} D'(1,2)*R'(2) &= (S(2)*R'(2))*R'(2) \\ &= S(2)*(R'(2)*R'(2)) \\ &= S(2)*0 \end{aligned}$$

$$=S(2)$$

であるから $D'(1,2)*R'(2)$ を計算すれば $S(2)$ が求まる。

【0213】

一般に、 j を奇数として、

$$\begin{aligned} D'(1,j)*D'(2,j) &= (S(j)*R'(j)*R'(j+1))*(S(j)*R'(j)) \\ &= (S(j)*S(j))*(R'(j)*R'(j))*R'(j+1) \\ &= 0*0*R'(j+1) \\ &= R'(j+1) \end{aligned}$$

であるから $D'(1,j)*D'(2,j)$ を計算すれば $R'(j+1)$ が求まる。

【0214】

また同様に、

$$\begin{aligned} D'(1,j+1)*D'(2,j+1) &= (S(j+1)*R'(j+1))*(S(j+1)*R'(j)*R'(j+1)) \\ &= (S(j+1)*S(j+1))*(R'(j+1)*R'(j+1))*R'(j) \\ &= 0*0*R'(j) \\ &= R'(j) \end{aligned}$$

であるから $D'(1,j+1)*D'(2,j+1)$ を計算すれば $R'(j)$ が求まる。

【0215】

この $R'(j)$ 、 $R'(j+1)$ を使用して $S(j)$ 、 $S(j+1)$ を求める。

【0216】

$$\begin{aligned} D'(2,j)*R'(j) &= (S(j)*R'(j))*R'(j) \\ &= S(j)*(R'(j)*R'(j)) \\ &= S(j)*0 \\ &= S(j) \end{aligned}$$

であるから $D'(2,j)*R'(j)$ を計算すれば $S(j)$ が求まる。

【0217】

また同様に、

$$\begin{aligned} D'(1,j+1)*R'(j+1) &= (S(j+1)*R'(j+1))*R'(j+1) \\ &= S(j+1)*(R'(j+1)*R'(j+1)) \\ &= S(j+1)*0 \\ &= S(j+1) \end{aligned}$$

であるから $D'(1,j+1)*R'(j+1)$ を計算すれば $S(j+1)$ が求まる。

【0218】

以上、乱数書き換え方式により再分割データを生成した場合には、3つの再分割データ $D'(1)$ 、 $D'(2)$ 、 $D'(3)$ のすべてを用いなくても、3つの再分割データのうち、2つの再分割データを用いて上述したように元データを復元することができる。

【0219】

また、乱数書き換え方式においても、一旦元データを復元することなく（元データが見える形で現れない）、データの再分割処理を行うことができるので、よりセキュアなデータ管理が可能となる。

【0220】

(3. 動作)

次に、図13を用いて、本実施の形態に係るデータ秘匿装置2の動作を説明する。

【0221】

まず、乱数生成部22が、記憶部21に記憶されたシード α をもとに擬似乱数生成アルゴリズムFにより、データSと同じ長さの乱数F(α)（上述した秘密分散法Aでの乱数Rに相当）を生成する（ステップS110）。

【0222】

次に、データSと乱数F(α)とから、秘密分散法Aを用いて、分割データD(1)、D(2)、D(3)を生成し、記憶部21に記憶するとともに、保管サーバ3a及び3bに保管する（ステップS120、S130）。

【0223】

尚、擬似乱数生成アルゴリズムFは、その時点（データSを擬似乱数アルゴリズムFを用いて分割する時点）において、コンピュータの計算能力からブロック暗号が解読不能であると判断される程度の計算量を要するアルゴリズムで、分割データ生成時にはデータ秘匿装置2に与えられている。

【0224】

時間の経過とともにコンピュータの性能が向上し、擬似乱数アルゴリズムFを構成するブロック暗号の脆弱化が予測された場合（例えば、具体的には、ブロック暗号解読に必要な計算量と、その時点で入手可能なコンピュータの計算能力の双方を鑑みて、数年程度で解読可能な場合など）には、解読に必要な計算量がより大きいブロック暗号を構成要素に持つ擬似乱数アルゴリズムGを用いて、記憶部21に記憶されたシード β をもとに、データSと同じ長さの乱数G(β)（上述した秘密分散法Aでの乱数R'に相当）を生成する（ステップS140、S150）。

【0225】

次に、記憶部21及び保管サーバ3a、3bに保管された分割データD(1)、D(2)、D(3)から秘密分散法Aを用いて再分割データD'(1)、D'(2)、D'(3)を生成し、分割データD(1)、D(2)、D(3)に代えて、記憶部21及び保管サーバ3a、3bに保管する（ステップS160、S170）。

【0226】

尚、擬似乱数生成アルゴリズムGは、上述したように擬似乱数生成アルゴリズムFより計算量が必要とされ、その時点（分割データを擬似乱数アルゴリズムGを用いて再分割する時点）において、コンピュータの計算能力からブロック暗号が解読不能であると判断される程度の計算量を要するアルゴリズムで、再分割データ生成時にはデータ秘匿装置2に与えられている。例えば、具体的には、擬似乱数生成アルゴリズムFの構成要素には128ビット鍵のAES（Advanced Encryption Standard）が用いられていた場合には、擬似乱数生成アルゴリズムGの構成要素としては、256ビット鍵のAESを用いるものである。

【0227】

以降は、上記ステップS140～S170の繰り返しである。即ち、時間の経過とともにコンピュータの性能が向上し、擬似乱数アルゴリズムGを構成するブロック暗号の脆弱化が予測された場合、解読に必要な計算量がより大きいブロック暗号を構成要素に持つ擬似乱数アルゴリズムHを用いて、記憶部21に記憶されたシード γ をもとに、乱数H(γ)を生成し、記憶部21及び保管サーバ3a、3bに保管された再分割データD'(1)、D'(2)、D'(3)から、秘密分散法Aを用いて、再分割データD''(1)、D''(2)、D''(3)を生成し、再分割データD'(1)、D'(2)、D'(3)に代えて、記憶部21及び保管サーバ3a、3bに保管する（ステップS140、S150、S160、S170）。

【0228】

尚、データSを復号するときは、記憶部21及び保管サーバ3a、3bに保管された分割データ（再分割データ）のうち、いずれか2つから秘密分散法Aを用いて復号化する（ステップS180）。

【0229】

従って、本実施の形態に係るデータ秘匿装置2によれば、ブロック暗号を用いた擬似乱数生成アルゴリズムにより生成された乱数を用いて、秘密分散法Aにより生成された分割データを、所定の時期にさらに、秘密分散法Aにより、より計算量の大きい擬似乱数生成アルゴリズムにより生成された乱数を用いて、さらに再分割するので、長期間にわたってデータを秘匿し続けることができる。

【0230】

特に、秘密分散法Aは、データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割方法であり、データを処理単位ビット長毎に区分けして、複数の元部分データを生成し、この複数の元部分データの各々に対応して、データのビッ

ト長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成して、所望の分割数の分割データを生成するとともに、新たに発生させた乱数から処理単位ビット長の複数の乱数部分データを生成し、各分割部分データと該乱数部分データの排他的論理和により処理単位ビット長毎に再分割部分データを生成して、所望の分割数の再分割データを生成するので、データを復元することなく、データを再分割することができるので、データをよりセキュアに管理することができる。

【0231】

尚、本実施の形態における秘密分散法Aは、多項式演算・剰余演算などを含む多倍長整数の演算処理を必要としないので、大容量データを多数処理する場合においても簡単かつ迅速にデータの分割および復元を行うことができるという効果を得ることができる。

【図面の簡単な説明】

【0232】

【図1】本発明の第1の実施の形態に係るデータ秘匿装置の概略構成を示すブロック図である。

【図2】本発明の第1の実施の形態に係るデータ秘匿装置の動作を示すフローチャート図である。

【図3】本発明の第2の実施の形態に係るデータ秘匿装置の概略構成を示すブロック図である。

【図4】秘密分散法Aの分割数 $n=3$ の場合の分割処理を示すフローチャートである。

【図5】秘密分散法Aにおいて16ビットの元データSを8ビットの処理単位ビット長に基づいて分割数 $n=3$ で3分割する場合の各データと定義式および各分割部分データから元データを復元する場合の計算式などを示す表である。

【図6】秘密分散法Aの分割数 $n=3$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式を示す表である。

【図7】秘密分散法Aの分割数が n で処理単位ビット長が b である場合の一般的な分割処理を示すフローチャートである。

【図8】秘密分散法Aの分割数 $n=3$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式の別の例を示す表である。

【図9】秘密分散法Aのデータ再分割処理（乱数追加注入方式）を示すフローチャートである。

【図10】乱数追加注入方式により元データSを元データSの半分の長さの処理単位ビット長に基づいて分割数 $n=3$ で再分割する場合の各データと定義式および各分割部分データから元データを復元する場合の計算式などを示す表である。

【図11】秘密分散法Aのデータ再分割処理（乱数書き換え方式）を示すフローチャートである。

【図12】乱数書き換え方式により元データSを元データSの半分の長さの処理単位ビット長に基づいて分割数 $n=3$ で再分割する場合の各データと定義式および各分割部分データから元データを復元する場合の計算式などを示す表である。

【図13】本発明の第2の実施の形態に係るデータ秘匿装置の動作を示すフローチャート図である。

【図14】バーナム暗号の一例を説明する図である。

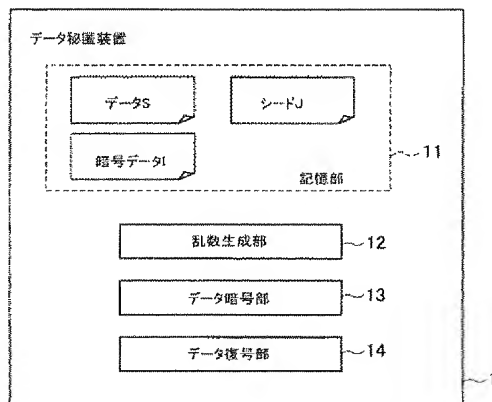
【符号の説明】

【0233】

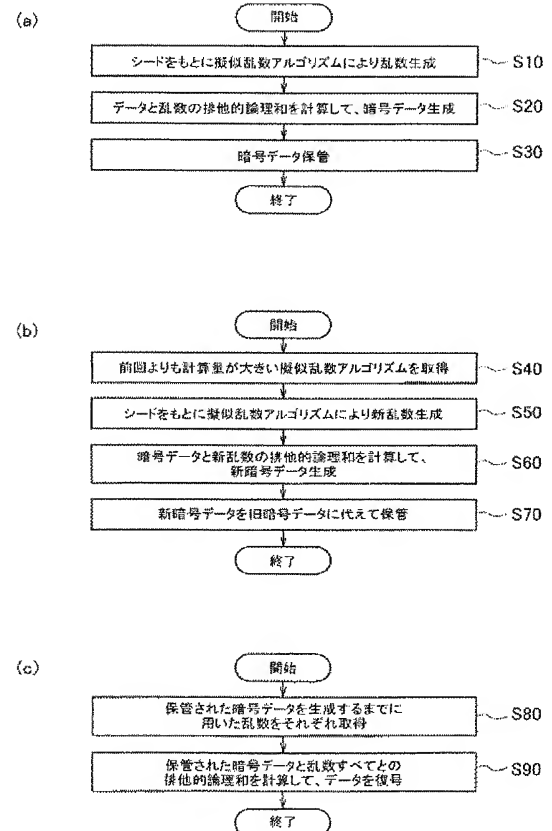
- 1, 2…データ秘匿装置
- 3 a, 3 b…保管サーバ
- 4…通信ネットワーク
- 1 1, 2 1…記憶部
- 1 2, 2 2…乱数生成部

- 1 3…データ暗号部
- 1 4…データ復号部
- 2 3…分割データ生成部
- 2 4…再分割データ生成部
- 2 5…元データ復元部
- 2 6…通信部

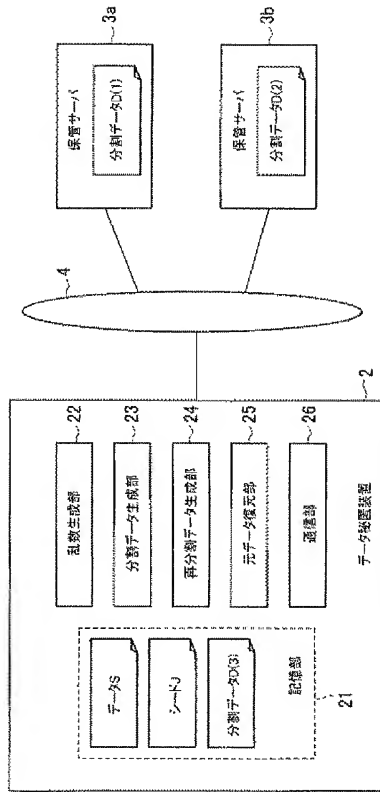
【図1】



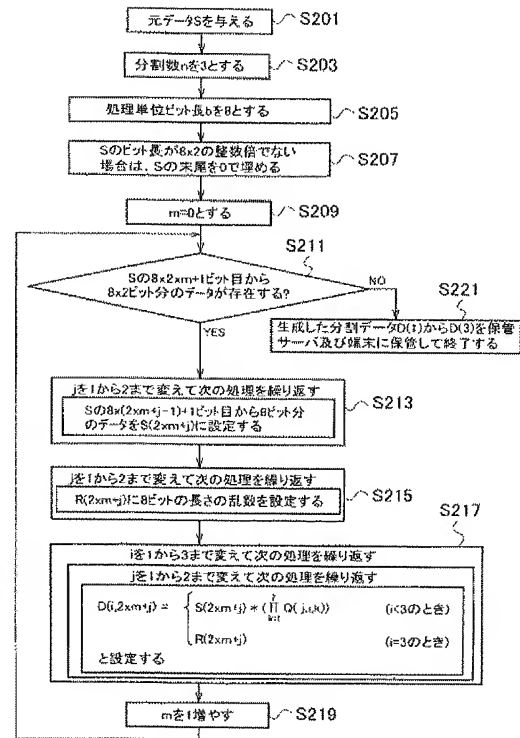
【図2】



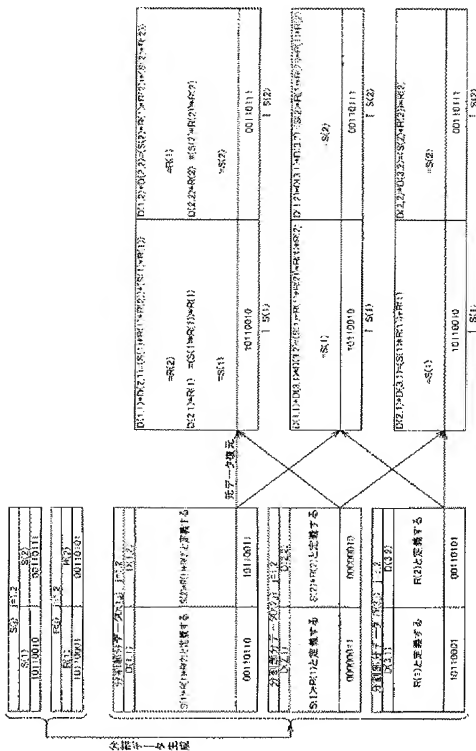
【図3】



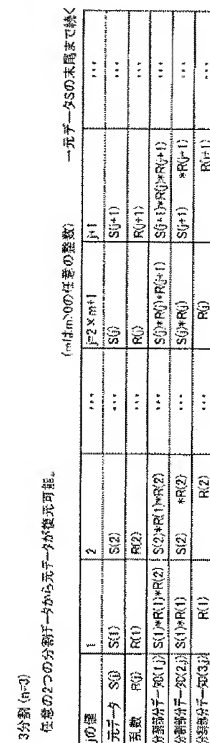
【図4】



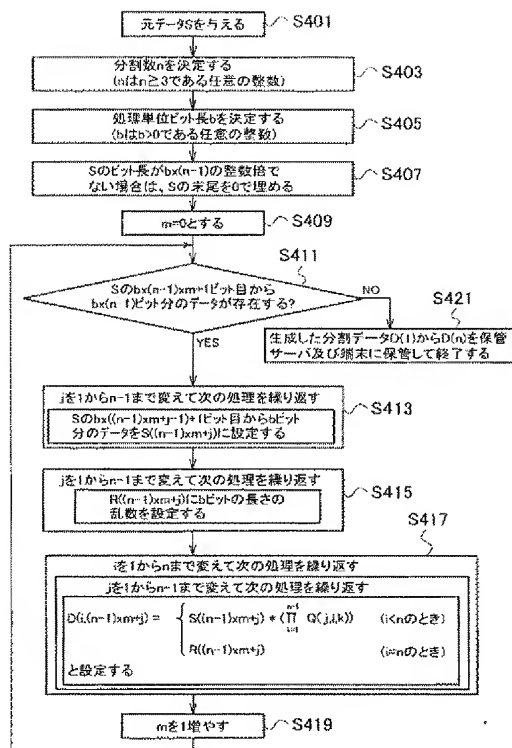
【図5】



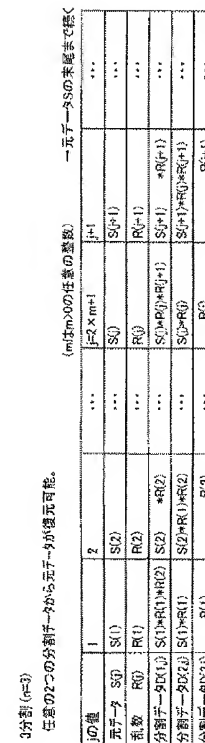
【図6】



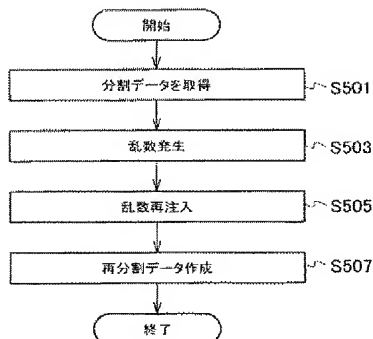
【図7】



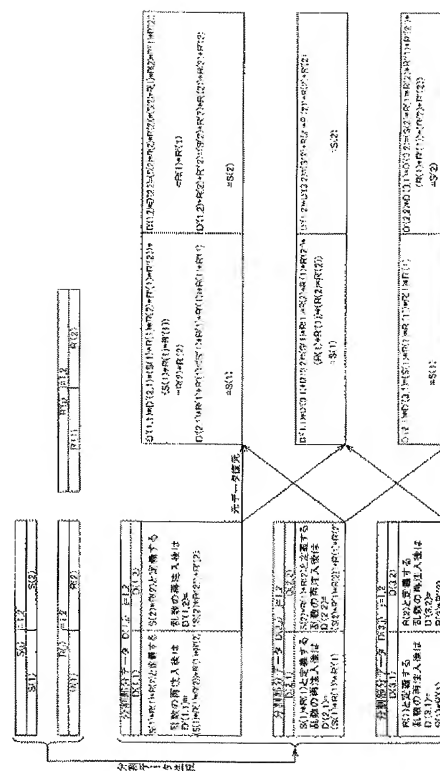
【図8】



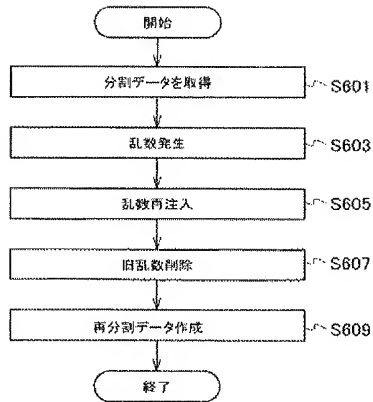
【図9】



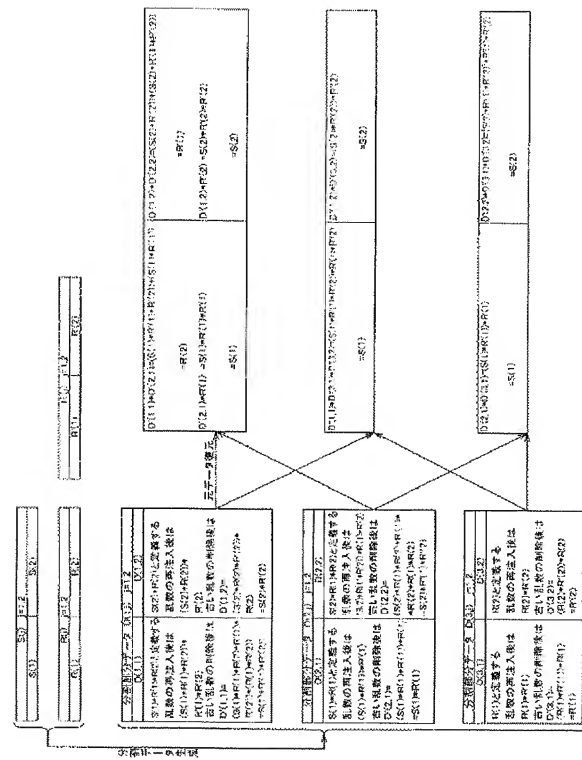
【図10】



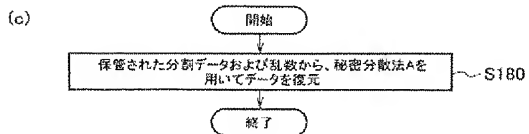
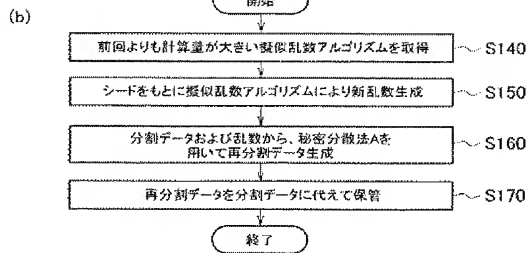
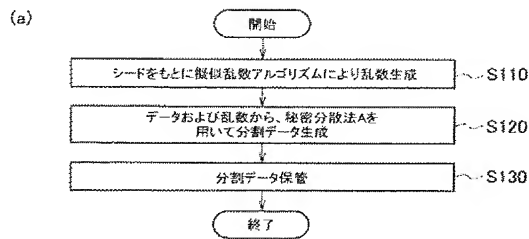
【図11】



【図12】



【図13】



【図14】

暗号化

1001001	1000110	明文X
1010110	0110001	鍵Y
0011111	1110110	暗号文Z

復号化

0011111	1110110	暗号文Z
1010110	0110001	鍵Y
1001001	1000110	明文X

- (72)発明者 荻原 利彦
東京都千代田区内幸町一丁目1番6号 エヌ・ティ・ティ・コミュニケーションズ株式会社内
- (72)発明者 加賀谷 誠
東京都千代田区内幸町一丁目1番6号 エヌ・ティ・ティ・コミュニケーションズ株式会社内
- (72)発明者 野村 進
東京都千代田区内幸町一丁目1番6号 エヌ・ティ・ティ・コミュニケーションズ株式会社内
- Fターム(参考) 5J104 EA13 FA01 JA04